STATISTICAL MODELS OF INTELLIGENT VIDEO-CONTENT ANALYSIS FOR COGNITION

by

Joseph Santarcangelo
Bachelor of Engineering, Ryerson University, 2007
Master of Engineering, Ryerson University, 2009

A dissertation
presented to Ryerson University
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
in the Program of
Electrical and Computer Engineering

Toronto, Ontario, Canada, 2017

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this dissertation. This is a true copy of the dissertation, including any required final revisions, as accepted by my examiners. I authorize Ryerson University to lend this dissertation to other institutions or individuals for the purpose of scholarly research. I further authorize Ryerson University to reproduce this dissertation by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my dissertation may be made electronically available to the public.

Statistical Models of Intelligent Video-Content Analysis for Cognition

JOSEPH SANTARCANGELO

Electrical and Computer Engineering, Ryerson University 2017

Abstract

Video content has a pronounced and varied cognitive impact. This thesis develops several statistical models of video and demonstrates that these models can be used as a means of quantifying how video impacts cognition. This work takes two approaches. For children, systems are developed to classify content based on expert recommendation. The second approach can be applied to adults and works by developing methods to determine extreme ranges of emotions that impact cognition. This thesis first develops decision fusion methods for cognitive classification of children's video content. It then introduces the novel concept of positive developmental classification of videos for children into videos that are deemed to have a negative or positive impact on cognition from a literature review; a novel system was developed to classify and segment the content accordingly. This study also introduces automatic age-based classification. The work focuses specifically on several high-level audio features as they relate to the cognitive capacity of children. As the impact on cognition of adults is dependent on the intensity of emotions, there is a focus on affective ranking. The main contributions include developing a method to rank and cluster sequences based on their affective content without the granularity problem. Furthermore, this thesis compares the accuracy of several regression methods on the LIRIS database and develops a method to incorporate prior knowledge into the cluster assignments. Then several state-based methods to predict valence and arousal are developed. The first method is the dynamic prediction-hidden Markov model for arousal-time curve estimation in sports videos. This method determines the arousal-time curve by selecting a state sequence that maximizes the joint probability density function between the arousal states and the arousal-time curve. The second method is a novel kernel-based mixture of experts model for linear regression. The latter method outperforms other mixtures of experts models in predicting valence and arousal. As the use of animation as a means of obtaining childrenâs attention, this thesis introduces a method to automatically categorize different animation genres in a video database made for children by statistically modelling the temporal texture attributes of the video.

Acknowledgments

There are many people to thank for helping make this thesis possible. First, I would like to thank the Ryerson University Department of Electrical and Computer Engineering for giving me the opportunity to start this thesis and conduct the necessary research work that I so much enjoy. I would also like to acknowledge the Department's helpful environment.

I would also like to thank my PhD supervisor, Prof. Xiao-Ping Zhang, for his advice, knowledge, enthusiasm and encouragement. He helped me to select a topic that allowed me the freedom to explore new ideas. I am also grateful to the department professors for their kind assistance. In particular, Prof. Ling Guan and Dr. Yifeng for all their help in reviewing my thesis and for their mentoring and encouragement.

A big thanks goes out to my student colleagues, Hui Zha, Triloke Rajbhandary, Iris Choi, Arsalan Bahojb, Timothy Little, Luan Vo, Newaz Rahim, Feifei Chen, Jianan Han, Jie Luo, Prathap Siddavaatam, Maryam Nematollahi and Richa Siddavaatam.

I feel a deep sense of gratitude to my family and to Ernest Della Penna for their encouragement and support.

Contents

List of Tables

List of Figures

List of Important Symbols

$\mathcal{D}$ data set

$h(\cdot)$ hypothesis.

Γ annotation.

$\mathbf{x}$ feature.

$f(\cdot)$ function

ζ noise

$P(\ldots)$ joint probability distribution

$\mathcal{L}(\cdot)$ loss

$R(\cdot)$ risk

$\mathcal{D}_T$ training set

$\mathcal{D}_V$ validation set

θ parameters

$\tilde{l}(\cdot)$ log likelihood

z independent latent variable

$\boldsymbol{\mu}$ mean

q_t hidden Markov latent variable

$\mathbf{o}_t$ observation variable

$b_i(\mathbf{o}_t)$ emission distribution

$\mathbf{w}$ vector of parameters for prediction

w_j $j-th$ parameters for prediction

ξ Gaussian distributed noise

σ standard deviation

$\boldsymbol{\phi}(\cdot)$ basis function

Φ data matrix of transformed data

α_p regularization term : lambda in the statistics literature

a dual variable

$\kappa(\cdot)$ mercer kernel

$\mathbf{K}$ gram matrix

$\mathbf{VC}$ VC dimension

ς Margin

Π Normalization matrix for clustering

Z Clustering assignment matrix

H Matrix used in relaxed solution

$h_i(\cdot)$ Decision Unit

S_f Set of features

S_t set of pixels that comprise a $t-th$ frame

$F_{n,t}$ set of pixels that comprise the $n-th$ face

$(\check{x}_{n,t}, \check{y}_{n,t})$ are the starting points of the bounding boxes containing the $n-th$ detected face

$(\check{x}_{w,n,t}, \check{y}_{h,n,t})$ dimensions of bounding boxes containing the $n-th$ detected face

$T_s(\tau)$ shot location

$\check{\lambda}$ parameter of the exponential function

X_l be the feature matrix of the $l-th$ episode

χ_m represent the feature matrix of $m-th$ series

γ_v validation set of series v

γ_{train} training set of series v

$\mathbf{Y}_i$ audio samples used for speech detection

LC language complexity

$\Delta STE(k)$ Short time energy

$\hat{\mathbf{y}}$ estimated point on the valence and arousal space

$\hat{Y}_i$ estimated sequence on the valence and arousal space

$\mathcal{K}(\cdot)$ dynamic time alignment kernel

$\mathbf{F}_l$ $l-th$ linking transform

$\hat{\mathbf{K}}_l$ $l-th$ linking transformed kernel matrix

$\mathcal{C}_1$ linking constant

R^2 squared correlation coefficient

Q^2 predictive leave-out squared correlation coefficient

R_n correct ranking label

C_m Cluster membership

$C_{m,n}$ assigned label of each $n-th$ video sequence samples

S_j the index set of the $j-th$ series

ACS_j The accuracy of series j

RSE The Residual squared error

s_j Arousal states

$y_{t,j}$ estimated arousal of a subject produced by video frame

$\hat{\pi}_i$ initial state distribution

$\hat{a}_{ij}$ transition distribution

A matrix of transition distribution

$\mathbf{a}$ vector of duel variables

$\mathbf{A}$ matrix of duel variables

μ_j mean of emission distributions

Q^* Viterbi state sequence

$\mathbf{\Lambda}_k$ indicator matrix

$\mathbf{v}_k$ parameters for gating probability density function

w_m mixture model coefficients

$Im(\cdot)$ gray level intensity array image

$\ddot{X}, \ddot{Y}$ dimensions of gray level intensity array image

$\Delta\ddot{y}, \ddot{x} \in Z$ displacement of intensity values

Chapter 1

Introduction

The problem of concern within this thesis is modelling and classifying the interaction between an individual's cognition and the audio and visual component of video content. Video is a means of conveying information using a combination of audio and visual inputs. With the rapid technological advances in digital TV, multimedia, and Internet, we have seen an amazing increase in video content. Individuals spend on average almost six hours watching television every day for entertainment purposes [3]. This duration increases for children [4]. In addition to entertainment, video content is also used as a learning resource in schools [5, 6], and employers are increasingly using online video content as a training resource [7, 8].

It has been shown video content impacts cognition [9, 10], and video with strong emotional content has an increased cognitive impact [10, 11, 12, 13, 14, 15, 16, 17]. Even more pronounced is the impact of video content on children. Video content can have a long-term negative impact upon grades, memory, and behavior [15, 16, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28]. An automatic method for classifying video content based on its effects on cognition would be extremely important and confer far-reaching benefits in many areas. We will refer to the audio and video properties that have some correlation with cognition as the cognitive contents of the video.

The problem will be decomposed into three sub-problems. The first will be classifying and predicting the impact of video content on cognition using expert labels, for example automatically classifying content using expert labels as set out in Chapters 3 and 6. As certain ranges of emotion impact cognition we will focus on ranking video time series based on emotional content as set out

in Chapter 4 and focus on affective modelling and determining states that may impact cognition as in Chapter 5. Finally, we will focus on easy-to-classify events or objects that the literature suggests may have an impact on cognition. This approach will be referred to as simple semantics.

The connection between chapters is shown in Fig. 1.1. The inner nodes represent the three components of the problem, and the outer nodes represent each chapter's connection. Emulating expert recommendation as in Chapters 3 and 6 is indicated in blue. Classifying simple semantics as in Chapters 3, 5, and 6 is indicated by the green nodes. Affective analysis, in particular ranking sequences to determine if they contain any extreme ranges of emotions that impact cognition, is indicated in red. Another related problem is determining states that are known to impact cognition. These states that could better model valence and arousal are indicated by the red nodes.

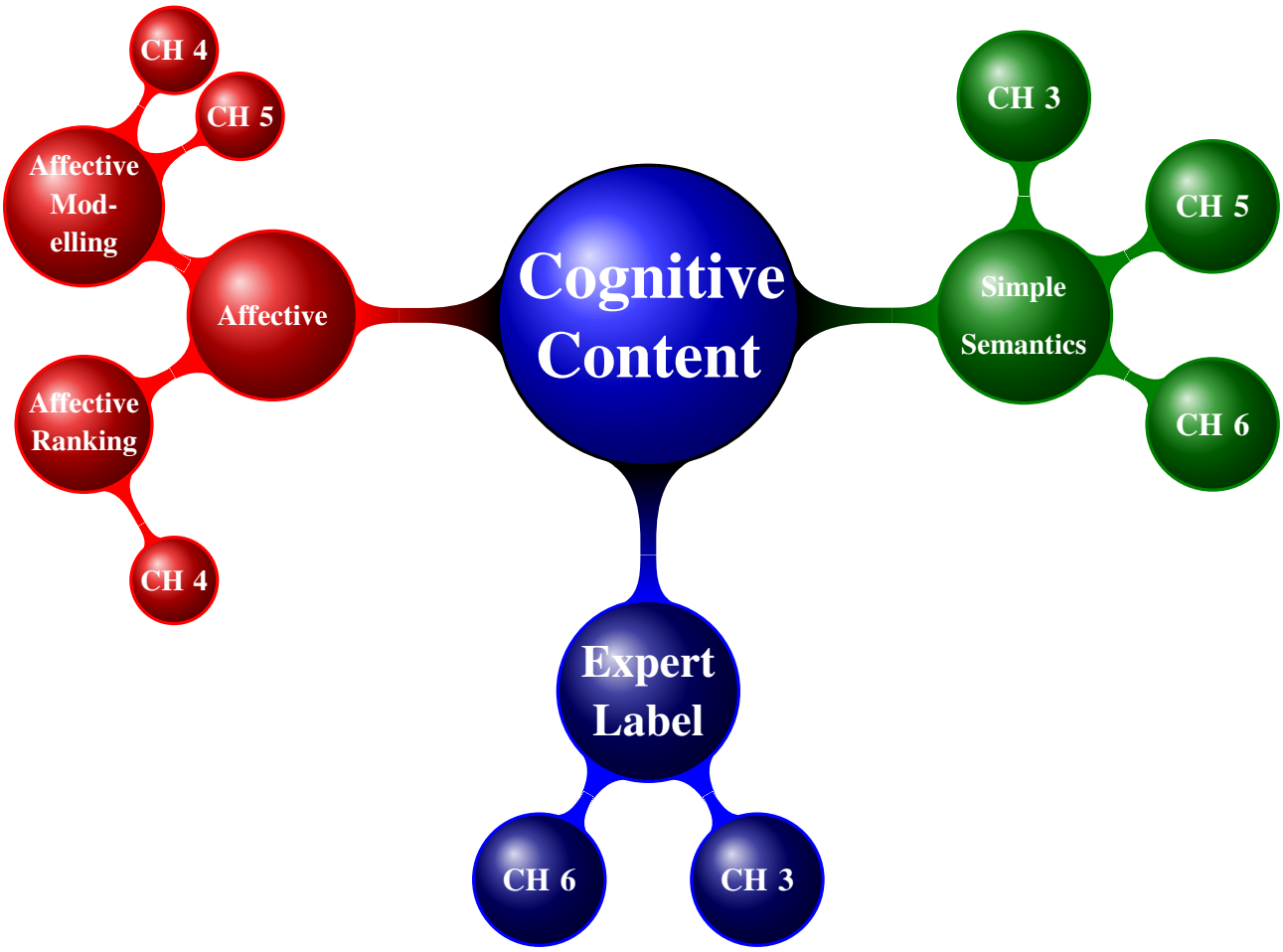


Figure 1.1: Breakdown of cognitive content and connection between chapters (CH).

1.0.1 What is Cognition?

Although this thesis is an engineering thesis, it is helpful to give a brief overview of what cognition is and how the term is used. Cognition is defined as the processes involved in the following [29]:

- Knowing
- Remembering
- Understanding
- Communicating
- Learning

A cognitive model describes a single cognitive phenomenon or process and describes the way humans process information. Cognitive models can be simple or complex, but what makes a good model is the ability to predict [29]. An example of a simple cognitive model from [30] is shown in Fig. 1.2. This model characterizes a response to an input. The yellow node represents cognition, the red node represents feelings and the green node represents action.

A good cognitive model must be emblematic of a particular cognitive process. The first attempts to create cognitive models in this thesis tried to construct a Bayesian cognitive model [31]. These methods did have some success [32], such as numerical advantages [33]. These models did not improve results in the latest data-sets [34] but did show better performance than models in the same class. In addition, as we are not concerned with the taxonomy of cognition, this thesis will try to derive models that better predict expert recommendations or cognitive responses to video content.

1.1 Background Work on Video-content Analysis

There are several problems associated with video-content analysis including shot boundary detection, key frame extraction, scene segmentation, extraction of features including static key frame

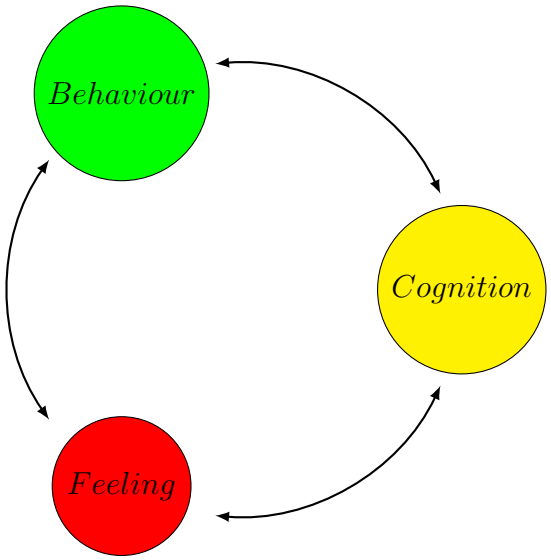


Figure 1.2: An example of a simple cognitive model used in cognitive therapy.

features, object features and motion features, video data-mining, video annotation, similarity measure, relevance feedback, video browsing, video retrieval, affective analysis and video classification [2, 35, 36]. To our knowledge, there has been no effort to classify video based on its cognitive content. The work here treats the problem in the context of video classification, but our approaches and methods can easily be extended to recommendation systems, expert systems, and video indexing. The next few sub-sections will give an overview of video classification and other multimedia fields that have been found to be useful in determining the cognitive content of a video sequence.

1.1.1 Video Classification

Considering the vast amount of video created every day, it is infeasible for individuals to watch and classify all these videos. The challenge is even more difficult when dealing with the cognitive content of video, as an expert is usually required to classify it. To overcome this problem, automatic video classification was created. Video classification assigns the videos into predefined categories.

TRECVid and MediaEval [37, 36] are two widely used video databases, and neither has explored the idea of classifying video based on its cognitive content. Much of the work in this thesis uses their methodologies. Although there are many similarities between video indexing [38] and

video classification we will focus on the video classification problem.

There are several different modalities used in video classification, which include social data, meta-data [39], text-based approaches, audio-based approaches, visual-based approaches, and various combinations of the aforementioned approaches [37, 36]. We will focus only on audio and visual approaches based on our original premise. The support vector machine has become the most popular method as feature sizes increase. Recently, deep networks have been incorporated into video classification, but this approach presents several issues. The latest work [40, 41] uses convolutional neural networks with feature pooling [42] or recurrent neural networks using Long Short Term Memory (LSTM) [43] to encode the frames. These methods were not applied as they require data sets as large as one million videos [40, 41].

Editing Effect Classification

Editing effect classification is not usually a main part of video content, but it is an important step. Several works use it as a first step in video classification [44, 35]. Although this work uses many motion features we will focus on scene changes. In this work most of the classification will be scene change detection or shot boundary detection, using classic histogram subtraction [45]. In the science literature, content with lots of scene changes is sometimes referred to as fast-paced. Content with lots of scene changes affects cognition [14]. This is especially true for children watching fast pace educational content [21], but the type of content also plays a role. For example educational programs with lots of scene changes have been shown to be beneficial [22, 23, 24]. As a result, we also review semantic content methods, such as video genre classification.

Semantic Classification

Video genre classification is an important part of determining the cognitive content. This problem has been well studied [35, 46, 36, 1, 47], and the particular genre is important in determining the impact like educational content [22, 23, 24], but genre is not the only factor. For example, different animation is used as a means of obtaining children's attention [28], yet the work in [18] found that some animation can be good for children while other animated content can be bad.

Other problems in video classification include finding the broadcaster in news video [46] and

finding sports video and events in sports video [48, 49, 50]. In [51] different sports videos were categorized using the fact that different sports genres have motion in different directions; the angle of motion field was compared to a prototypical set of motion vectors. In [52], a combination of HMM and support vector machines were used as a classifier; their feature space included a combination of color histogram moments and color coherence vectors [53]. Motion information was incorporated using frame differencing. Edge direction and edge intensity histogram information were also used. In [54], a similar approach was taken as in [52]. The main difference was the use of continuous observation densities HMM (CODHMM) [55]. Another computationally inexpensive feature is the Gray Level Co-occurrence Matrix (GLCM).

The first use of GLCM in video genre classification was [1]. In [1] a combination of colour, audio, cognitive and structural features was used with textural features. In [56], the block intensity comparison code (BICC) was developed. The BICC characterizes a frame based on each block. The feature vector was then reduced by principle component analysis and a CODHMM was used as the classifier. The BICC had better performance than any individual feature in [54] and proved that the gray level distribution of each block could be effective in discriminating between genres. The elements of BICC provide a measure of how similar a block is to every other block in one frame of a video sequence.

There are other aspects of video classification that have a relation to cognitive content; for example, violence and horror have been found to be bad for children [22, 57, 58] and automatic violence and horror scene detection has been performed by [59, 60, 61], but this thesis is different in that there is much content that affects cognition that is not violent and does not contain horror. As a result, the semantic problem only partially fulfills the requirements and a more direct relationship between the visual and audio content of a video sequence and the cognitive content must be found.

It should be noted that the term cognitive has been used before in video-content analysis [62], but the approach used in this work lies more in the context of semantic classification. Low-level features, as physical signal stimulations to our brains, may directly contribute to the physiological cognitive brain response. We therefore focus this thesis work on lower level features. It will be another interesting research topic to compare the different contributions to cognitive brain response

between low-level and high-level features. One helpful approach is that taken by affective video-content representation. In this work, we will use the term cognitive to describe methods directly related to cognition.

1.1.2 Affective Video Content Representation

Affective video content representation is the intensity and type of feeling or emotion elicited by a video. One of the first works on affective video-content representation was performed by [2]. Since then there has been a large body of work on this topic [63, 64, 65, 66, 34].

These publications demonstrate different methods and features used in affective video content representation that affect feelings and emotion. Furthermore, it has been shown that affective analysis can be used to detect violent and scary content [17, 60, 67, 68]. There is a large body of work showing the relationship between physiology, cognition and low-level features. Affective analysis is related to cognition [2] and several publications in the scientific literature show that the affective content of media play a role in processes more strongly associated with cognition, such as memory and attention [11, 10, 12, 13, 14, 15, 16].

The timeline in Fig. 1.3 helps illustrate the parallels between affective video content representation in the multimedia community compared to work in other areas collected within a literature review. The yellow circles are proportional to the number of publications found relating to the relationship between affective analysis and cognition and their dates. The red circles are proportional to connected publications in affective video content representation. It is evident that, in the last ten years the two fields have paralleled each other, but, to our knowledge, there has been no effort to combine the two areas.

Valence and Arousal Plane

The most direct representation of an emotion is to use discrete labels or emotional prototypes. Examples include fear, anxiety and joy. This method has many problems: labels are not universal, labels can be misinterpreted and emotions are continuous phenomena rather than discrete [64]. Finally, fixed classes can be changed only by combining or splitting certain classes to reduce or

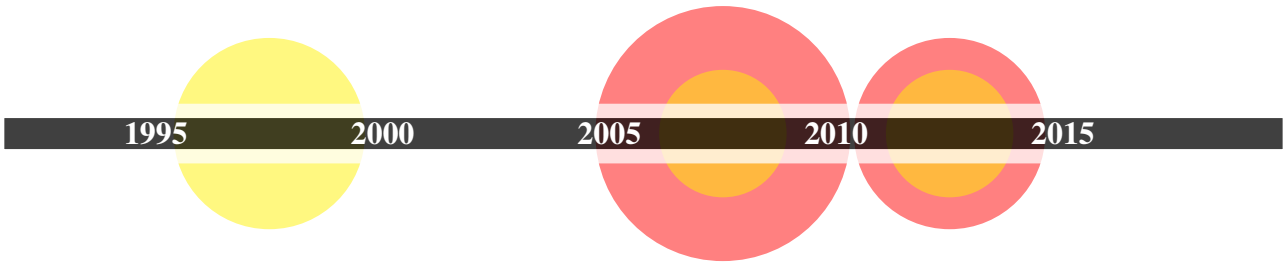


Figure 1.3: Yellow circles are proportional to the number of publications found relating to the relationship between affective analysis and cognition. Red circles are proportional to important publications that contain databases in affective video content representation

increase the emotional granularity [69, 70, 65, 66, 34].

Another method is to use the 2-D emotion space to describe the emotional content of a video sequence [2]. This space contains the valence dimension and the arousal dimension. Valence describes the type of emotions: negative to positive. Arousal describes the intensity: inactive to active. Any point on this space can be used to describe different emotions. It has been shown that low-level video features can be mapped onto this space using regression [66]. Much of the literature in the scientific community quantifies emotional content with respect to cognition with valence and arousal [11, 10, 12, 13, 14, 15, 16]. An example of the 2-D emotion space with some discrete labels is shown in Fig. 1.4. The curves represent emotion generated by two possible video sequences; the red sequence elicits negative feelings, such as fear, anger and calm. The second sequence generates positive feelings, such as happiness and relaxation. The curves are not necessarily closed but drawn like that for aesthetic purposes.

Cognition and the Valence and Arousal Plane

Usually, certain regions of the valence and arousal plane impact cognition [11, 10, 12, 13, 14, 15, 16, 13]. Regions with intense emotion can interfere with cognitive processing. It has also been demonstrated that moderate levels of emotion have been shown to improve performance in many aspects of processing including attention and memory. For example, regions that are high or low in arousal may have an adverse impact on attention, memory, and vigilance, while medium

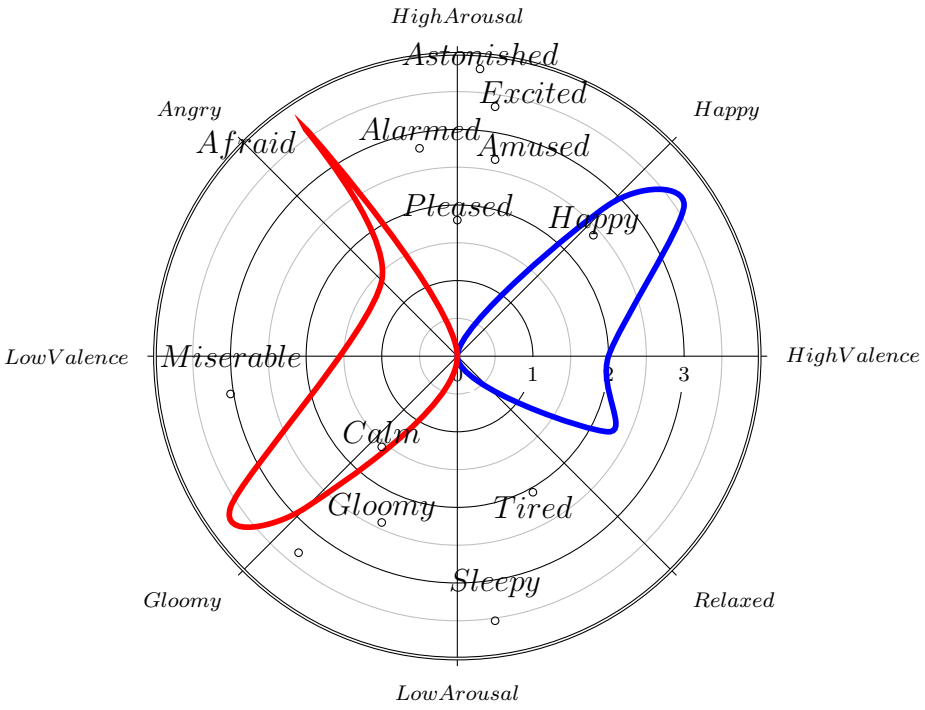


Figure 1.4: 2-D emotion space: Diagonal axis represents valence, horizontal axis represents arousal, in addition there are several discrete labels corresponding to different emotional prototypes and two curves represent generation by two possible video sequences [2].

regions may be positive. This is demonstrated in Fig. 1.5, where red represent regions that have a negative impact on cognition while green regions have a positive impact on cognition. As there exist annotations of arousal and valence for regression, these methods can be used to predict points on the plane [15, 16]. Then a relative ranking system can be used to determine the impact on cognition.

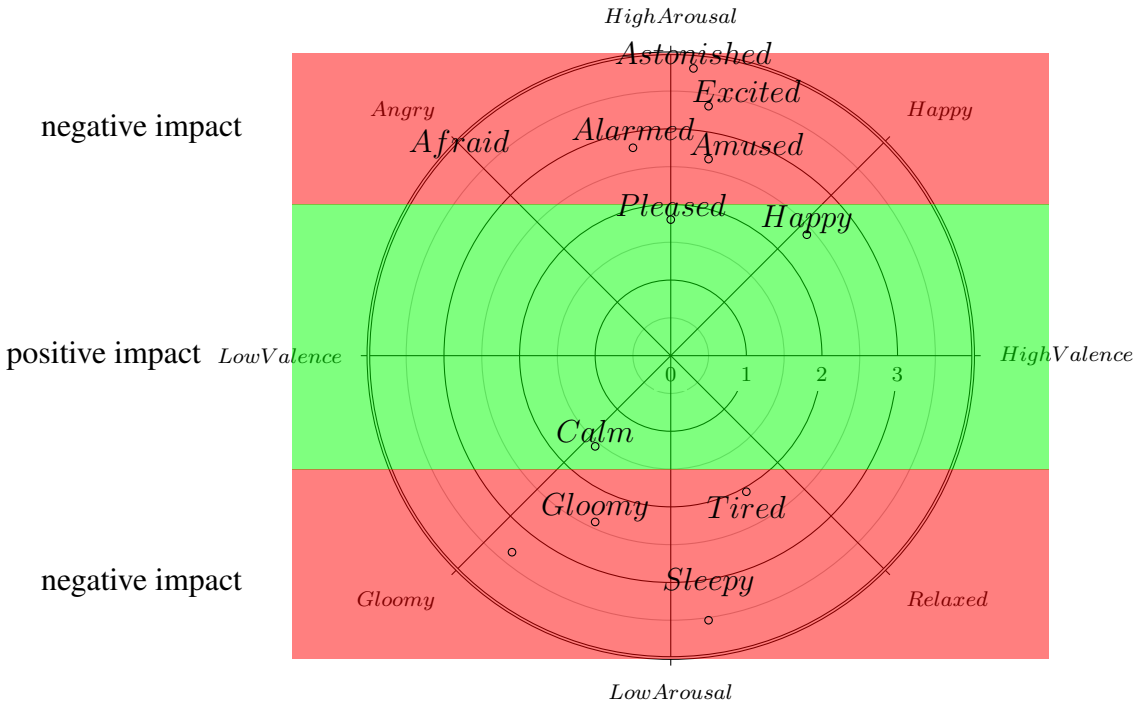


Figure 1.5: Valence and arousal plane: different colors represent regions of the VA space that have a positive impact (green) or a negative impact (red).

Affective Annotation

Another advantage of affective analysis is the wide range of available datasets. Although many of the labels are manually annotated, the correspondence between physiological response and manual labelling has been shown in [66]. There has been a wide body of work in affective data-sets. The HUMAINE [71] was designed to illustrate the concept of affective computing. RILMSTIM [72] was perhaps the first affective database that could be used with respect to this thesis, but the labels

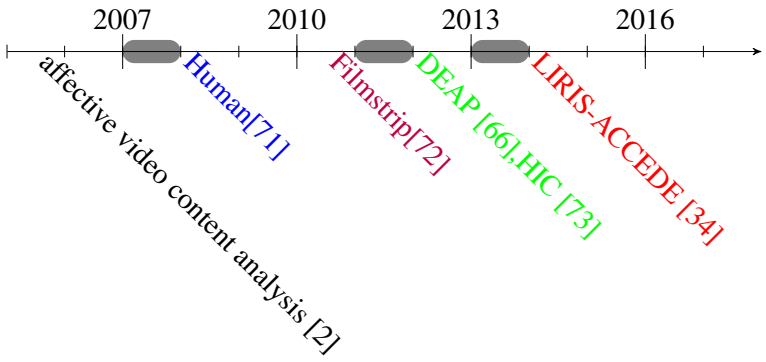


Figure 1.6: Timeline of some important data-sets related to one of the first publications on affective video content analysis.

are global, making it difficult to work with. The DEAP data-set [66] and HIC data-sets [73] are also available, but many of the links to [66] are missing and [73] only uses short segments of the clips. As a result, this work uses [34] as the videos and labels are easily obtained and a large number of subjects were used. Fig. 1.6 shows the timeline starting from one of the first works in affective video-content analysis [2] and demonstrates its rapid progression over the last five years.Valance and arousal are just intermediate steps in determining the cognitive content. Similar results can be obtained directly by analyzing video content features without computing the mapping to the valance arousal plane.

1.1.3 Not the Full Story

Although these methods quantify the emotional content of a video, they do not cover all the problems associated with cognitive content of a video sequence. The work in [12] shows that emotional content of media plays a role in attention, but so does interest, as well as several other factors. This concept is illustrated in Fig. 1.7. The large ellipse represents all the factors that could represent cognitive content. One important factor in cognition is attention represented by the blue ellipse. The work in [12] shows that emotional content affects attention, symbolized by the intersection with the affective content ellipse. The same work also suggests that the viewer's interest in the video also plays a role, indicated by the intersection of the interest ellipse. It is a much more

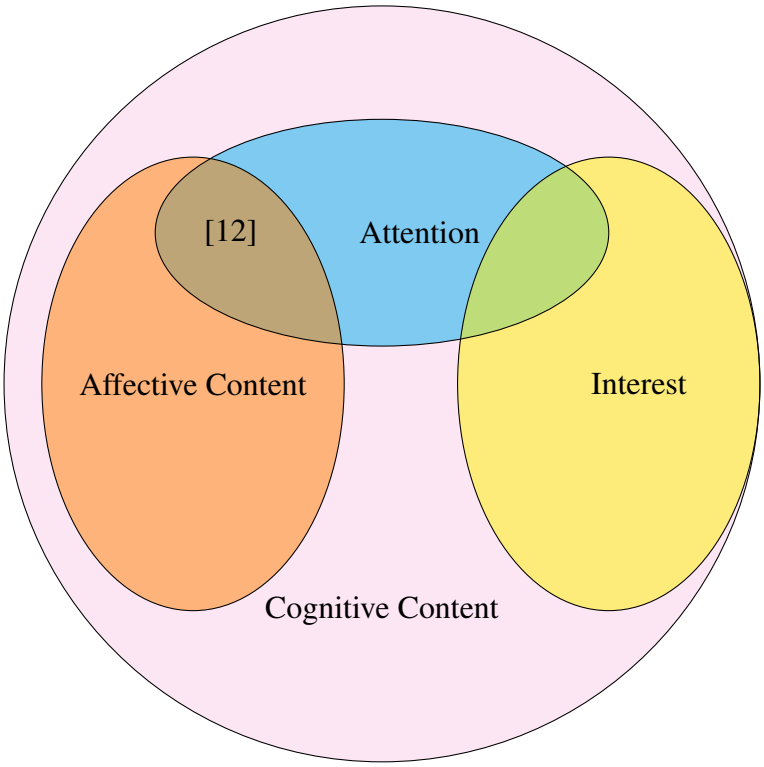


Figure 1.7: Large ellipse represents factors that could represent cognitive content: Affective content represented by the orange ellipse, interest ellipse in yellow and attention represented by the blue ellipse. The intersection with the interest ellipse represents that both factors impact attention.

difficult problem to quantify the user's interest in a video using video and audio features, yet the relationship between affective video and audio features seems to be established. Therefore, in addition to combining classification and semantics, other methodologies must be employed.

1.1.4 Other Features

Many of the features used in video-content analysis are included in other areas in multimedia, most notably image and audio-content analysis. In this section, we include other features and their relationships. Music affects cognition [74]; as a result, we use features from music retrieval [75]. Many of these features are similar to those used in affective analysis [36, 1, 47, 2, 76] and similar features can be used to recognize the emotional state in the human voice [77], such

as aggression. Dialogue-heavy content is usually good and voiced speech can be detected using spectral features [78]. The structure of language is important thus we use concepts from Natural Language Processing (NLP) [79].

The Mind Map in Fig. 1.8 gives a high-level summary of factors that have an impact on cognition and the relationship to features used in the multimedia community using the references from previous sections. The orange nodes represent areas of study in the multimedia analysis community.

The green nodes indicate factors that have been shown to have a positive impact, and factors that usually have a negative impact, especially on children, are shown in red. Yellow nodes represent factors that, to our knowledge, have not been studied. The edges that connect the nodes represent some relationship between the areas of interest. For example, affective analysis used audio, color, and intra-frame techniques.

1.1.5 Cognitive Content of a Video Sequence

It is not difficult to see that the cognitive content of a video sequence can have many applications. In this work, we develop several models to determine the cognitive content of a video sequence. The models are developed using predefined expert assigned classes; methods and features are determined using recommendations from expert observations and selected if they minimize generalization error.

Since the impact on video content is so important to children we focus on automatically classifying children's video based on pre-defined cognitive categories. The database was collected by reading the scientific literature and with the pre-defined cognitive categories assigned by experts in the same literature. Feature selection and engineering was also determined using the same literature and verified with model validation techniques [15, 16, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28]. As there is no database for children's pre-defined cognitive categories, the database was collected by reading the scientific literature. The pre-defined cognitive categories assigned to each series was determined by what the experts recommended in the literature. Most of the series in the novel database had at least two citations, the latest study was performed in 2011 [18].

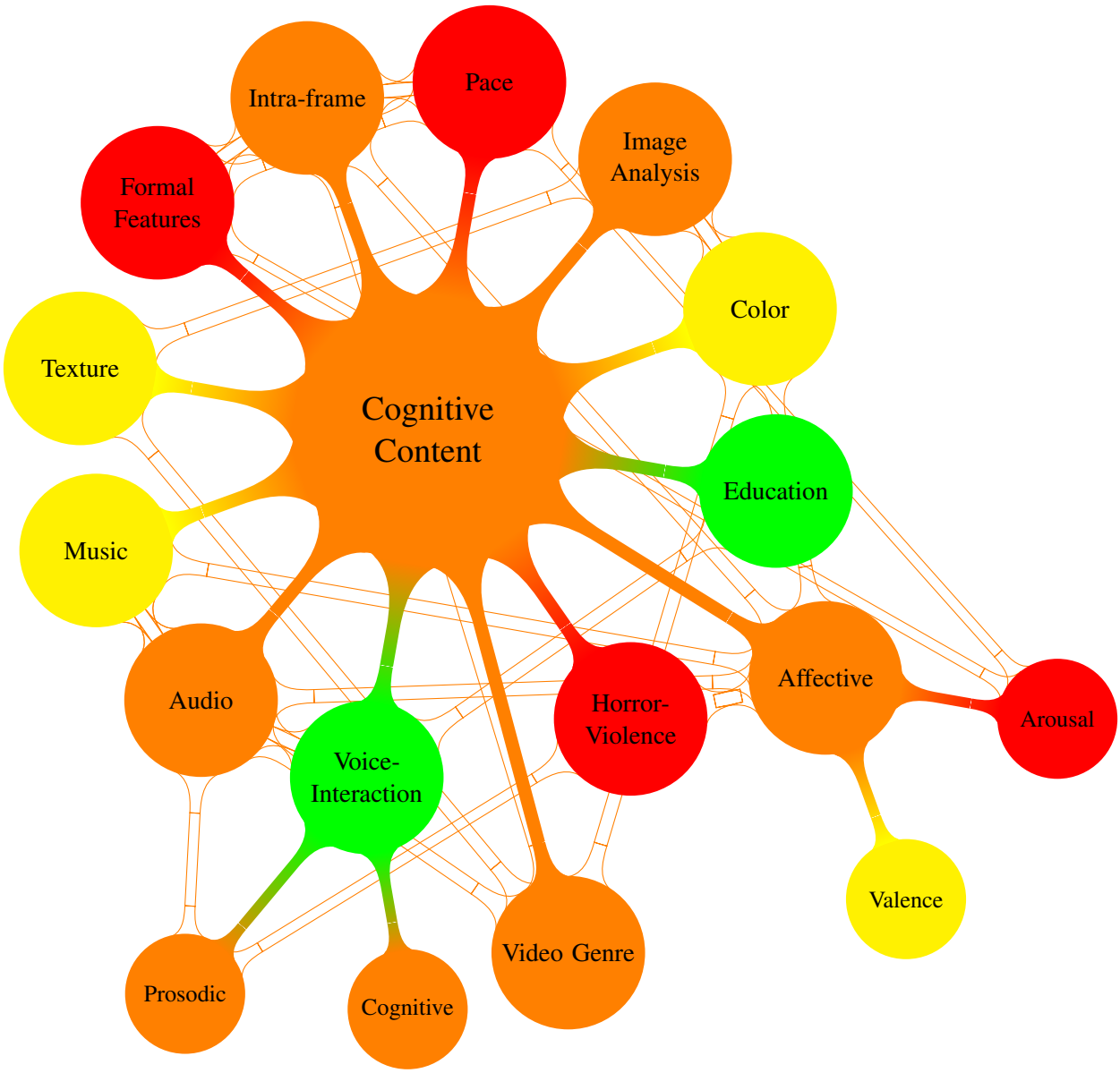


Figure 1.8: Mind map: Red nodes represent factors that have a negative impact, yellow nodes have a unknown impact and green has a positive impact. Orange nodes represent areas that have been studied in the multimedia community and edges represent relationships or common features.

Another task that is an ideal application of the cognitive content is determining the appropriate age of content for young children. The task is possible for content made for young children because the language in content for children of three years of age is so much different compared to the language in content for children of five or even four years of age. In addition, sounds that may interfere with the bandwidth associated with voice are minimal in this content. The method determines the average number of words and syllables and determines a measure of how complex the language is. This information is then used in combination with other information to determine the appropriate age of the target audience.

It has been shown that certain ranges of valence and arousal have a negative impact on adults and children's cognition [10, 11, 12, 13, 14, 15, 16]. As a result, we develop a method to rank entire video sequences on different components of the valence arousal plane. The method is unsupervised and does not face the granularity problem. For example, even if values are mapped onto the valence arousal plane the continuum has to be quantized again in order to produce possible rankings [69]. Furthermore, this method tests how different methods perform in mapping features on the valence arousal plane.

In addition, several novel regression methods were developed to predict the arousal or valence-time curve. The first method is the dynamic prediction HMM for arousal time curve estimation in sports videos. The method determines the arousal time curve by selecting a state sequence that maximizes the joint probability density function between the arousal states and the arousal time curve. The second is a novel kernel-based mixture of experts model for linear regression. The method outperforms other mixtures of experts models for predicting valence and arousal from the LIRIS database.

Finally, because different types of animation are used to gain children's attention [24, 25, 26] and imaginary characters mixed with live action improve attention [80], we develop a method to classify different types of animated content that perform better than more complex general methods for video genre classification.

Compared to other problems, the cognitive content problem can be viewed as a mid-level problem, though much can be accomplished using low-level features and affective analysis, such as

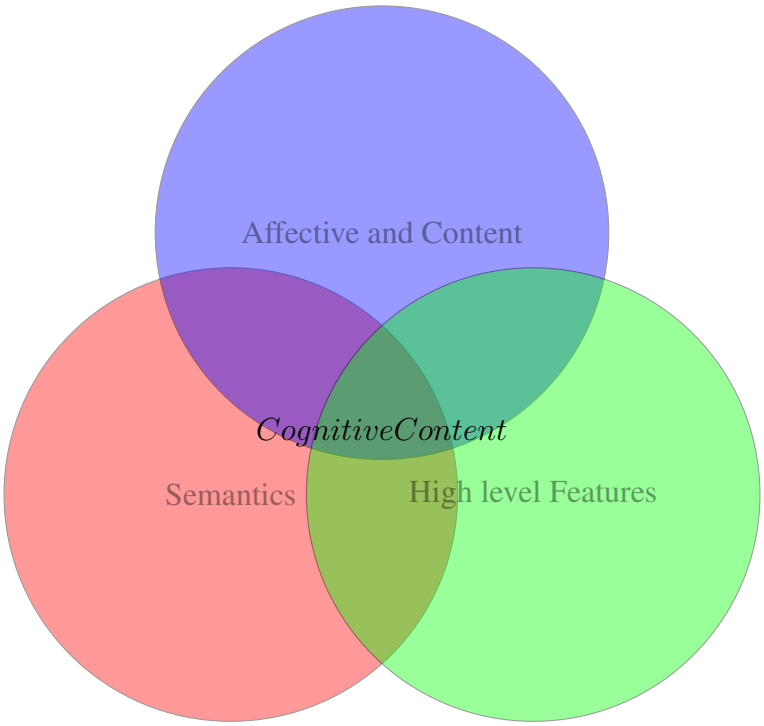


Figure 1.9: Venn Diagram showing that cognitive content is a combination of high level features, affective analysis and semantic classification.

classifying. If content has a negative or positive impact on children, the problem is still more complex. For example, using classification methods, such as determining the number of words or syllables, can be extremely useful in the age recommendation task. These simpler semantic tasks are high-level compared to low-level features or affective analysis directly in the classification task, but still not as complex as other semantic tasks. The diagram in Fig. 1.9 demonstrates the relationship between semantics, high level features, affective and content based features.

1.1.6 Connection Between Valence, Arousal and Children's Videos

It is helpful to look at the connections between chapters as they relate to the respective results and the work of Dr. Annie Lang. Dr. Lang's work pre-dated and influenced much of the present work done on affective computing and is used to explain why some content has a negative or positive impact on cognition [17, 14, 10, 14, 11, 12, 81, 14]. Consider Fig. 1.10, the red nodes represent each chapter and the directed edges represent the influences between them. The connection be-

tween Chapters 3 and 6 involves determining factors that impact children's cognition hence the connection between nodes. Also in Chapter 3 it was found that features that correspond to high arousal and certain ranges of valence have a negative impact on cognition. As a result Chapters 4 and 5 focus on ranking arousal and/or valence locally and globally as indicated by the edges between nodes. This relationship was predicted in the literature represented by the outer nodes in Fig. 1.10. The green nodes represent publications by Dr. Annie Lang [17]. This work was directly cited by [20] in purple, whose work showed that the number of scene changes is correlated with cognitive impact on children. This paper along with several that cited it was key for much of the work in Chapter 3 denoted by the directed edges in Fig. 1.10. In addition, Dr. Lang's work was directly cited in [2], as represented by the link from the green nodes to the blue nodes. This was used in Chapters 3, 4 and 5. In addition, these features have been found to be extremely accurate in predicting whether content had a negative or positive impact on cognition.

1.1.7 Thesis Outline

The remainder of this thesis is organized as follows: Chapter 2 introduces the problem in terms of empirical risk minimization and introduces the preliminaries; Chapter 3 discusses positive developmental video classification for children and introduces an automatic age recommendation system for children; Chapter 4 elaborates on the ranking of video sequences using the valence arousal plane; Chapter 5 develops a dynamic prediction hidden Markov models for arousal time curve estimation in sports videos and a Kernel-based mixture of experts for valence arousal estimation. Finally, Chapter 6 discusses animation genre discrimination, as summarized in Fig. 1.11.

Fig. 1.12 shows the connections between each chapter. All chapters are based on observations in the scientific literature, as symbolized by the parent node. The children of the node marked *Expert Assigned* represent chapters that use some pre-defined expert labels. The children of the node marked as *Affective* represent chapters that use affective analysis. Similarly, the children of the node marked *Classification* are chapters that use some kind of classification based on an expert recommendation that impacts cognition. The recommendation may be used to create a high-level features, such as counting the number of words, or for a more general task, such as determining

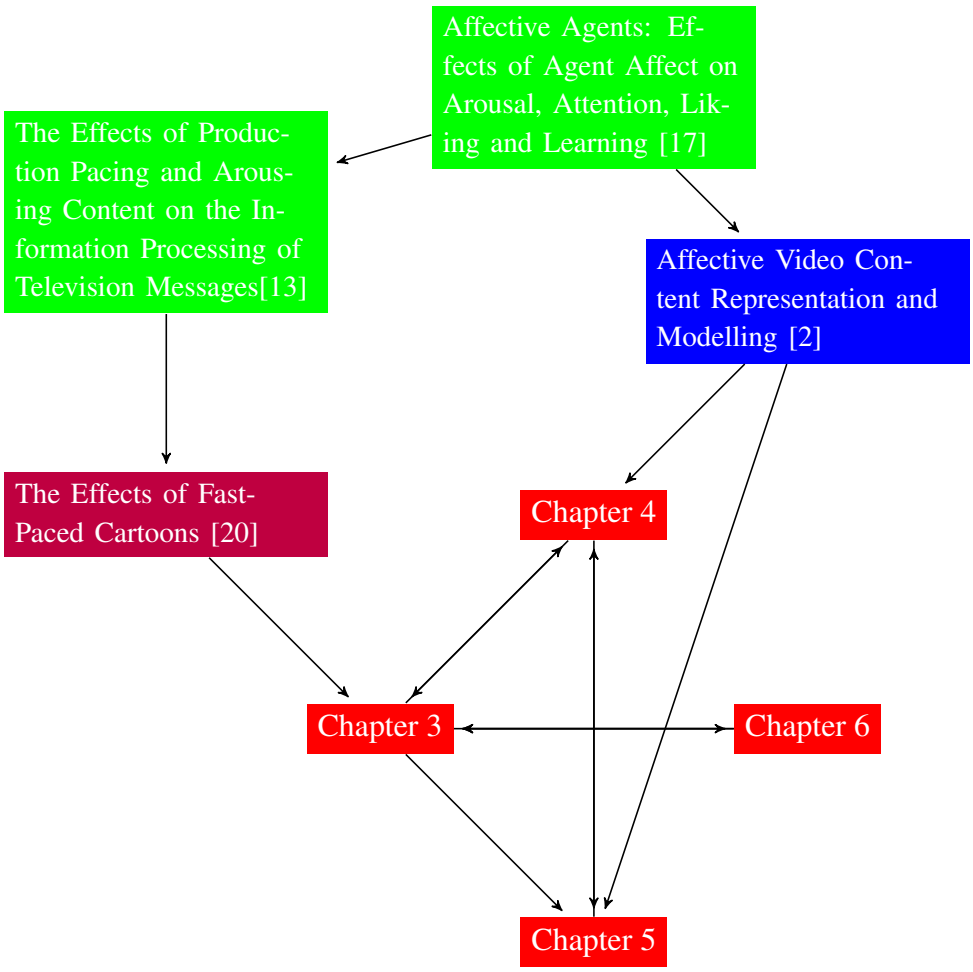


Figure 1.10: Connection between chapters and literature.

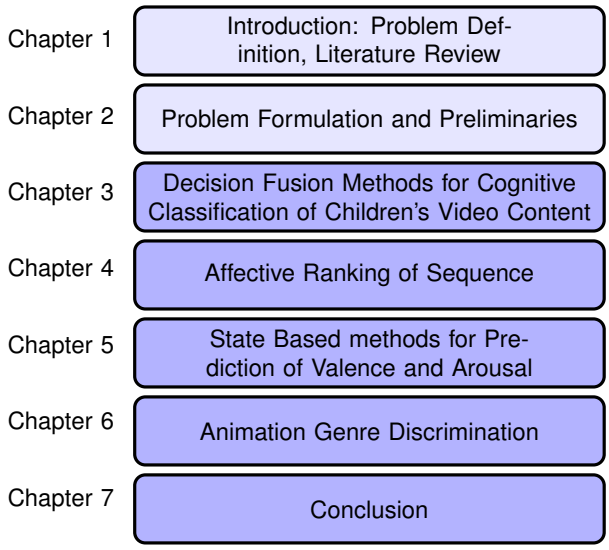


Figure 1.11: Thesis Outline: Light blue represents introduction of concept and review; dark sections represent novel models.

genre.

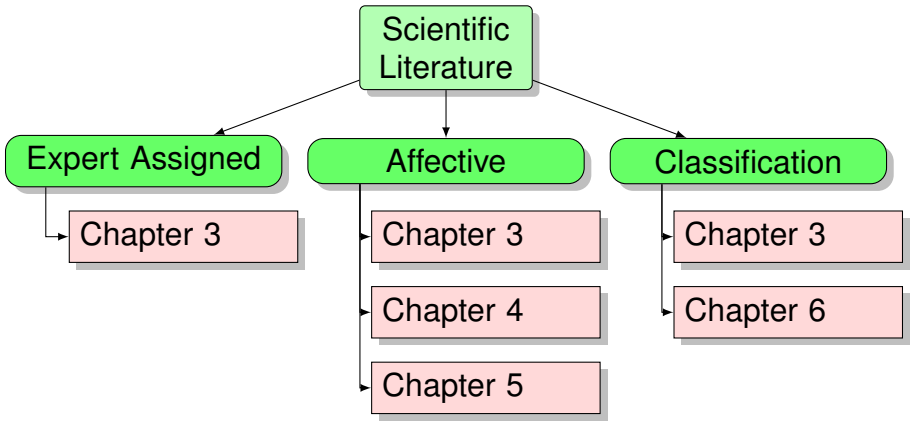


Figure 1.12: Connection between chapters.

1.1.8 Main Contributions

The main contribution of this thesis is the introduction of the concept of cognitive content of a video sequence. Other contributions are set out in this section. In Chapter 3, we introduce the novel concept of positive developmental video classification for children. In addition to the novel

research topic, we collect a set of videos that have been deemed as having a negative or positive impact on child cognition. From a literature review, a novel model validation technique was developed. Several new features and several experiments on how sampling rate affects classification were conducted and novel methods were used to segment clips using categorical clustering. Then we introduce automatic age-based classification. In addition to a novel research topic, we introduce several novel high-level audio features related to the cognitive capacity of children. These novel features gauge the cognitive ability of the intended audience by quantifying the structure of the language. These novel features include syllable rate, word rate, language complexity, and noise jumps. The feature extraction methods are also novel in that we count the number of syllables and words using relatively computationally inexpensive signal-processing techniques, foregoing complex speech recognition. Given the relationship between emotions and cognition, Chapter 4 develops a method to rank sequences using their affective content without the granularity problem. In addition, Chapter 4 compares the accuracy of several regression methods on the LIRIS database and performs regularization and variable selection via the Elastic Net and Lasso methods. Chapter 5 develops a dynamic prediction- hidden Markov model for arousal-time curve estimation in sports videos and a kernel-based mixture of experts for linear regression. The dynamic prediction-hidden Markov model determines the arousal-time curve by selecting a state sequence that maximizes the joint probability density function between the states and the arousal-time curve. We derive the parameters using the expected maximization algorithm. Experiments were performed on several types of sports videos including golf, bowling, darts, and tennis. Test measures included squared residual error and criteria derived from psychology. The experimental results show that the novel method performed better in estimating the arousal-time curve than state-of-the-art linear regression methods for most of the tested sports videos. A kernel-based mixture of experts was also developed using the dual formulation to constrain the maximum likelihood estimation. This method outperforms other mixtures of expert models and has comparable performance to other methods for regression. Finally, due to the use of animation as a means of obtaining children's attention, we introduce a method to automatically categorize different animation genres in a video database made for children. There has been research in animation genre categorization [82], but the method

developed here does not use colour allowing it to classify older black and white content. This method is based on statistically modelling the temporal texture attributes of the video sequence.

Chapter 2

Preliminaries

As this is an engineering dissertation we will formulate the problem in terms of empirical risk minimization [83]. Consider the following situation, which is a general setting of many supervised learning problems. Given some data set $\mathcal{D}$, one would like to determine some hypothesis h that minimizes the loss between some expert annotation Γ and some hypothesis $h(\mathbf{x}|\mathcal{D})$, where $\mathbf{x}$ is some feature. The observation Γ is actually a noisy version of some function $f(\mathbf{x})$ for example $\Gamma = f(\mathbf{x}) + \zeta$. More formally, we assume that there is a $P(\mathbf{x}, \Gamma)$ that is the joint probability distribution. This allows us to model uncertainty in predictions of Γ. We also assume that we are given a non-negative real-valued $\mathcal{L}(h(\mathbf{x}|\mathcal{D}), \Gamma)$ which measures how different the prediction $h(\mathbf{x}|\mathcal{D})$ of a hypothesis is from the true outcome Γ. The risk is given by:

$$R(h) = \mathbf{E}[\mathcal{L}(h(x|\mathcal{D}), \Gamma)] = \int \mathcal{L}(h(\mathbf{x}|\mathcal{D}), \Gamma)\, P(\mathbf{x}, \Gamma) d\mathbf{x} d\Gamma. \tag{2.1}$$

One would like to minimize equation 2.1, but the joint probability function is difficult to determine. Using the law of large numbers, we can approximate the true risk and determine the best hypothesis using the following:

$$h^* = \underset{h \in \mathcal{H}'}{\operatorname{argmin}} \{ \frac{1}{N} \sum_{n=0}^{N} \mathcal{L}(h(\mathbf{x_n}|\mathcal{D}), \Gamma_n)) \}. \tag{2.2}$$

Where N is the number of samples and $\mathcal{H}$ is the hypothesis space. In some cases sequences will be classified so the notation $\mathcal{L}(h(\mathbf{X}_n), \Gamma_n)$ will be used, where $\mathbf{X}$ is a matrix representing a sequence. For parametric models $\mathcal{D} = \theta$, one can also define the problem in terms of utility function, whose value we would like to maximize. For each problem we will define the risk or utility function.

The main problem with this formulation is it does not account for the fact that features can be the output of a hypothesis. In the cognitive content problem consider the age classification system for young children, the language is indicative of the appropriate age of the content. The complexity of the language is one factor that can be used to quantify the language and the complexity of the language is related to the ratio of words to syllables.

As a result, we train a classifier to determine the appropriate age by using the ratio of words to syllables as a feature. To automatically segment and count the number of words and syllables we use a classifier. Counting the number of syllables was based on detecting voiced speech which also can be posed as an empirical risk minimization problem. Each step is an independent problem requiring its own training and classification. Each level assumes a dependency on the previous step and the order is determined by domain knowledge of psychology and multimedia content analysis. As a result of this multi-level paradigm, we use recursion to simplify the notation and pose the problem in terms of empirical risk minimization and minimize it in a greedy fashion.

Problem Formulation

Let $S_0 = \{(\Gamma^{0,1}, \mathbf{x}^{0,1}), ..., (\Gamma^{0,N_{f0}}, \mathbf{x}^{0,N_{f0}})\}$ be the features with an associated target each with its own training set. For the feature and target $(\Gamma^{0,j}, \mathbf{x}^{0,j})$, the training set of samples is:

$$D^{0,j} = \{(\mathbf{x}^{0,j,0}, \Gamma^{0,j,0}), ..., (\Gamma^{0,j,N_{f0j}}, \mathbf{x}^{0,j,N_{f0j}})\}. \tag{2.3}$$

The features and targets will be used to train a set of classifiers $\{h^{0,1}, ..., h^{0,N_{f0}}\}$. In the initialization we will perform an empirical risk minimization for each hypothesis. The initialization step is defined as:

$$h^{0,j} = \underset{h \in \mathcal{H}^{0,j}}{\operatorname{argmin}} \{ \frac{1}{N_{f0j}} \sum_{n=1}^{N_{f0j}} \mathcal{L}_{0,j}(h(\mathbf{x}^{0,j,n}), \Gamma^{0,j,n})\}. \tag{2.4}$$

In the $l-th$ step we define the set of features as $S_l = \{(\Gamma^{l,1}, \mathbf{x}^{l,1}), ..., (\Gamma^{l,N_{flj}}, \mathbf{x}^{l,N_{flj}})\}$. Where $\mathbf{x}^{l,1} = h^{l-1,1}(\mathbf{x}^{l-1,1}), ..., \mathbf{x}^{l,N_{fl}} = h^{l-1,N_{fl-1}}(\mathbf{x}^{l-1,N_{fl}})$ and the rest of the values are new features.

The induction step is defined as:

$$h^{l,j} = \underset{h \in \mathcal{H}^{l,j}}{\operatorname{argmin}} \{ \frac{1}{N_{flj}} \sum_{n=1}^{N_{flj}} \mathcal{L}_{l,j}(h(\mathbf{x}^{l,j,n}), \Gamma^{l,j,n}) \}. \tag{2.5}$$

In the final step L, we will drop the superscript and sub-scripts feature vector and target, and the minimization is given by: $(\mathbf{x})_1 = h^{L-1,1}(\mathbf{x}^{L-1,1}), ..., (\mathbf{x})_{L-1} = h^{L-1,1}(\mathbf{x}^{L-1,1})$.

$$h^{L} = \underset{h \in \mathcal{H}}{\operatorname{argmin}} \{ \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}(h(\mathbf{x}_n, \Gamma^n) \}. \tag{2.6}$$

Example

Consider a simplified version of Chapter 4. It is known that certain features are correlated with valence and arousal denoted by: $S_0 = \{(\Gamma^{0,0}, \mathbf{x}^{0,0}), (\Gamma^{0,1}, \mathbf{x}^{0,1})\}$ where $(\Gamma^{0,0}, \mathbf{x}^{0,0})$ are the valence targets and valence features and $(\Gamma^{0,1}, \mathbf{x}^{0,1})$ are the arousal targets and arousal features. The loss function is given by:

$$h^{0,j} = \underset{\mathbf{w}_{0,j} \in \Re^{d_{0,j}}}{\operatorname{argmin}} \{ \frac{1}{N_{f0j}} \sum_{n=1}^{N_{f0j}} (\Gamma^{0,j,n} - \mathbf{w}_{0,j}^T \boldsymbol{\phi}(\mathbf{x_n}))^2 \} \tag{2.7}$$

The new values are outputs on the valence arousal plane $\mathbf{x} = [\mathbf{x}^{1,0}||\mathbf{x}^{1,1}]^T$ and are correlated with the valence and arousal rankings. Each sample is a member of a sequence and can be ranked using clustering. The clustering step can be viewed as a parameter estimation problem [83] more precisely maximum likelihood estimation. The cluster parameters can be determined by minimizing:

$$h^{2} = \underset{\theta}{\operatorname{argmin}} \{ \frac{-1}{N} \sum_{n=1}^{N} ln(P(\mathbf{x}_n|\theta) \}. \tag{2.8}$$

2.1 Model Selection

Cross validation [84] is a means of approximating generalization error and determining optimal models and/or model parameters that cannot be optimized directly. The first and most important distinction is between training and validation data. Training data is the data used to train the model

and is also referred to as in-sample data. The validation data is used to estimate prediction error for model selection. Finally, in many cases, free parameters are involved that are selected empirically; as a result, a test set is used for assessment of the generalization error of the final chosen model.

2.1.1 Cross Validation

Determining model performance using the training data would just validate the method's ability to fit the training data, but the method may fail to predict anything useful on unseen data. Consider a data set:

$$\mathcal{D} = \{(\mathbf{x}_1, \Gamma_1), .., (\mathbf{x}_N, \Gamma_N)\} = \{(\mathbf{x}_n, \Gamma_n)\}_{n=1}^{N}. \tag{2.9}$$

Where $\mathbf{x}_n$ is a feature vector and Γ_n is a continuous value or a discrete label. The simplest form of cross validation is to randomly partition the data into two mutually exclusive sets; a training set $\mathcal{D}_T$ and the validation set $\mathcal{D}_V$, with corresponding indexes $IND(\mathcal{D}_V)$ and $IND(\mathcal{D}_T)$. A block diagram is shown in Fig. 2.1. The loss or error of the validation set is given by:

$$\mathcal{L}_V(h) = \frac{1}{N_V} \sum_{n \in \mathcal{D}_V} \mathcal{L}(h(\mathbf{x}_n | \mathcal{D}_T), \Gamma_n). \tag{2.10}$$

Where N_V indicates the number of samples in the validation set, expected error of the validation is given by $E(\mathcal{L}_V(h)) = \mathcal{L}_{out}(h)$. Assuming that the samples are independent, the variance is given by:

$$var(\mathcal{L}_V(h)) = \frac{1}{N_V^2} \sum_{n \in \mathcal{D}_V} var(\mathcal{L}(h(\mathbf{x}_n | \mathcal{D}_T), \mathbf{\Gamma}_n)) = \frac{\sigma_V^2}{N_V}. \tag{2.11}$$

If N_V is large we can use the normal distribution; the confidence of the estimate is bounded by the size of the validation set:

$$\mathcal{L}_V(h) = \mathcal{L}_{out}(h) \pm \sigma_V O\left(\frac{1}{\sqrt{N_V}}\right). \tag{2.12}$$

Therefore, as N_V gets larger, one can develop a better estimate of the range of $\mathcal{L}_V(h)$, but because the number of training examples is given by $N_T = N - N_V$, there is less training data for the algorithm. Depending on the number of samples and the complexity of the model, if N_V gets large the bias of the model may increase. As a result, K-fold cross-validation was developed to test

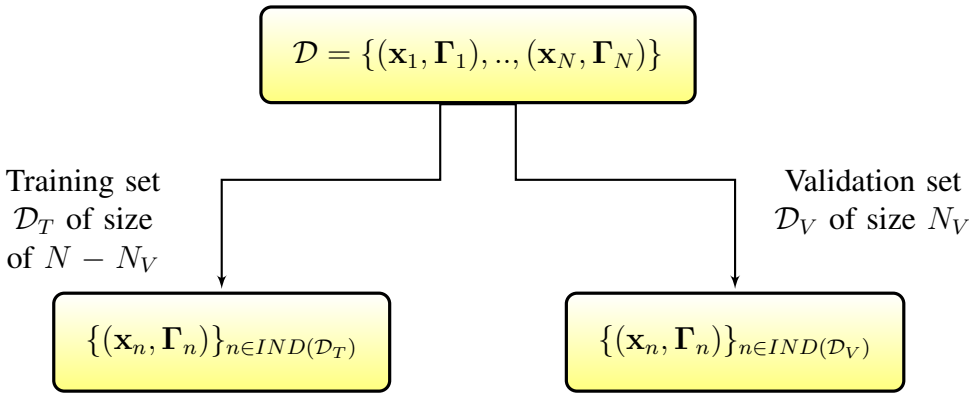


Figure 2.1: Partitioning data into training and validation data.

the model with different training and validation sets iteratively. The data is partitioned in such a way that no samples are used for both training and validation in one iteration.

2.1.2 K-fold Cross-Validation

K-fold cross-validation is a re-sampling procedure. This method works by dividing the data set into K_f subsets, then using some sub-sets for training and some sets for validation. The final loss is determined by averaging the results. The method works if there is a small amount of data and you would like to use every sample for training and testing or if there is a large amount of data and it is not feasible to use all the samples. Usually the data set is partitioned into K_f subsets and the procedure is repeated K_f times. As a result, one generates a validation set $\mathcal{D}_{V,k}$ for every iteration K_f times N_V. The data set can be decomposed as follows:

$$\mathcal{D} = \{\bigcup_{j=1}^{K_f} \mathcal{D}_{V,j} : \mathcal{D}_{V,i} \cup \mathcal{D}_{V,l} = \emptyset\}. \tag{2.13}$$

The expected loss of the validation data is given in 2.14, where the dependence of the hypothesis on the training data is made explicit:

$$\mathcal{L}_V(h) = \frac{1}{K_f} \sum_{j=1}^{K_f} \frac{1}{N_{V,j}} \sum_{n \in \mathcal{D}_{V,j}} \mathcal{L}(h(\mathbf{x}_n | \mathcal{D}_{T,j}), \mathbf{\Gamma}_n). \tag{2.14}$$

In addition, the loss of each fold is given by:

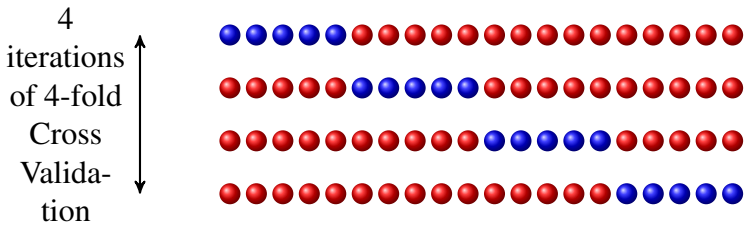


Figure 2.2: 4-fold cross-validation with three folds used for training; the red samples represent training data and the blue samples represent validation data.

$$\mathcal{L}_{V,j}(h) = \frac{1}{N_{V,j}} \sum_{n \in \mathcal{D}_{V,j}} \mathcal{L}(h(\mathbf{x}_n|\mathcal{D}_{T,j}), \mathbf{\Gamma}_n). \tag{2.15}$$

An example of four-fold cross-validation is shown in Fig. 2.2. Twenty samples are divided into four groups of five. The red samples represent training data and the blue samples are used for validation data. Three of the folds are used for training and the other fold is used for testing. The procedure is repeated four times until each set is used for validation once.

There is an interesting trade-off between how the folds are used and the variance of the estimation. The uncorrelated argument used in Eq. 2.11, is harder to justify when all the folds are used for testing. This is because the samples $\mathcal{D}_{T,j}$ are used in training multiple times. The variance can be expressed as:

$$var(\mathcal{L}_V(h)) = \frac{1}{K_f^2} \sum_{i=1}^{K_f} \sum_{j=1}^{K_f} cov(\mathcal{L}_{V,l}(h), \mathcal{L}_{V,j}(h)) = \frac{1}{K_f^2} (\sum_{i=1}^{K_f} \sigma_i^2 + (\sum_{j \neq i}^{K_f} cov(\mathcal{L}_{V,i}(h), \mathcal{L}_{V,j}(h)))). \tag{2.16}$$

We see that the variance of the estimation depends on the correlation of the loss of the folds. As a result, a larger K_f will have a smaller variance if the samples are uncorrelated, but if the samples are correlated, a large K_f may not decrease the variance of the results. Therefore, selecting a K_f is done heuristically. $K_f = N$ or leave-one-out cross-validation is popular as well as $K_f = 3$, $K_f = 6$ and $K_f = 10$. If it is not feasible to use all the training data, some of the folds may be left out.

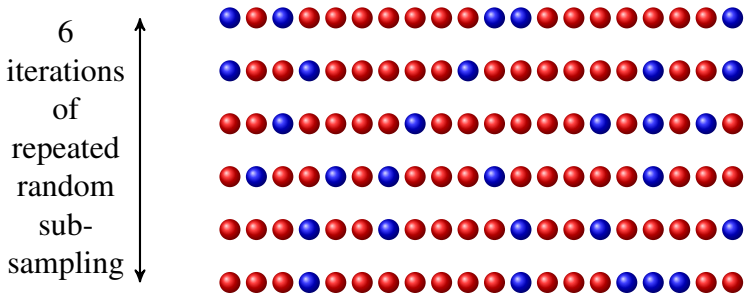


Figure 2.3: 6 iterations of repeated random sub-sampling; red represents training data and blue represents validation data.

2.1.3 Repeated Random Sub-Sampling Validation: Bootstrap

This method randomly splits the data-set into training and validation data; then the results are averaged. The main advantage of this method is that it is not limited by the number of the folds. The disadvantage is that some samples may not be used or some samples may be used multiple times. The main difference between this method and k-fold cross-validation is that condition 2.13 does not apply and $\mathcal{D}_{V,j}$ is randomly selected from $\mathcal{D}$ for every iteration. As a result, the number of iterations is not limited because different permutations from $\mathcal{D}$ can be generated. An example of six iterations is shown in Fig. 2.3, where blue represents validation data and red represents training data.

2.1.4 Test Data

In many cases, models have free parameters that are not directly determined by the training algorithm; for example, what kernel works better in a support vector machine, how many states in a HMM or how many mixtures in a mixture model. These values are usually selected using the validation data, but this is considered an optimistic estimate. As a result, once the optimum value for the model has been determined using the validation data, the final performance of the algorithm is determined using a separate set of samples called the test data given by $\mathcal{D}_{Test}$. This can be incorporated into all the above algorithms. A block diagram is shown in Fig. 2.4.

An example of four-fold cross-validation is shown in Fig. 2.5, where two of the folds are used

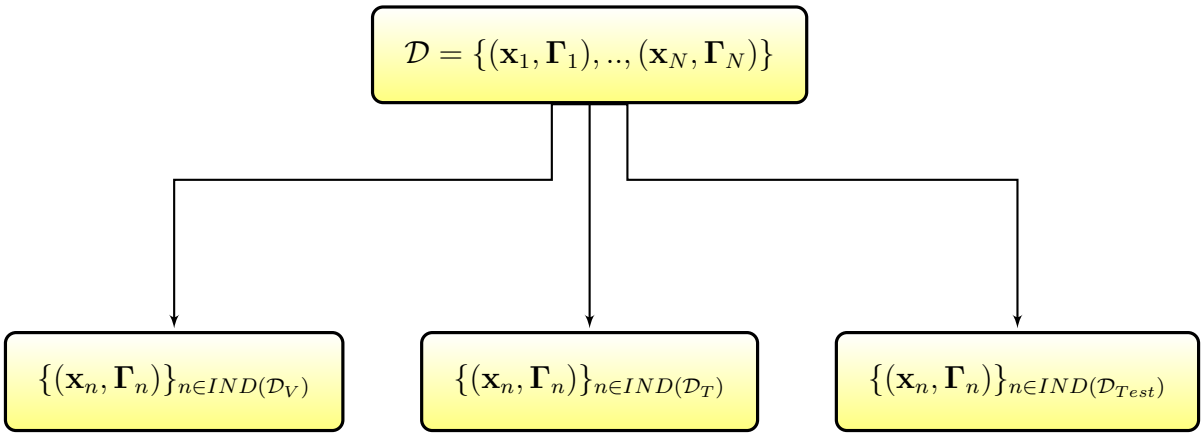


Figure 2.4: Partitioning data into training, validation and testing data.

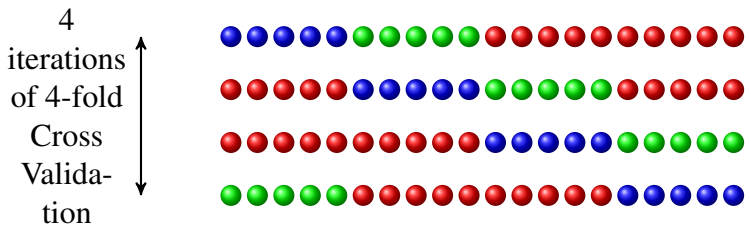


Figure 2.5: 4-fold cross-validation with three folds used for training. The red samples represent training data, the blue samples represent validation data and the green samples represent test data.

for training in red, one fold is used for validation in blue and the final fold is used for testing in green. The procedure can be modified for different variations of cross-validation.

2.2 Optimization and Estimation

2.2.1 Primal and Dual

This section is devoted to duality because it is not as intuitive as other optimization techniques and is widely used in Chapter 5. Duality is a method of constrained optimization that can be used to find the minimum lower bound of a function. The method transforms the initial problem into a problem that has an optimum solution [85]. Depending on the form of the problem, this optimum solution is the same or close to that of the original problem. We consider an optimization problem

in the standard form:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f_0(\mathbf{x}) \\ \text{subject to} \quad & f_i(\mathbf{x}) \leq 0, \quad i = 1, \ldots, m \\ & g_i(\mathbf{x}) = 0, \quad i = 1, \ldots, p. \end{aligned}$$

Minimizing the solution with respect to $\mathbf{x}$ directly is known as the primal problem. We will let p^* denote the primal solution as the minimum value of $\mathbf{x}$ in the regions that can satisfy the constrains. This is known as the feasible region: $\mathcal{D}_\wr = \bigcap_i^m \{dom(f_i)\} \bigcap \{\bigcap_i^p dom(g_i)\}$. Using a Lagrange multiplier we can convert the constrained problem into an unconstrained problem:

$$\Lambda(\mathbf{x}, \mathbf{a}, \boldsymbol{\nu}) = \left(f_0(x) + \sum_{i=1}^{m} a_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x) \right). \tag{2.17}$$

where $\nu_i > 0$.

The Karush Kuhn Tucker (KKT) conditions state that the constrained problem can be solved via the unconstrained problem.

We define the Lagrange dual function (or just dual function) as the minimum value of the Lagrangian function:

$$g(\mathbf{a}, \boldsymbol{\nu}) = \inf_{x \in \mathcal{D}_\wr} \Lambda(\mathbf{x}, \mathbf{a}, \boldsymbol{\nu}) = \inf_{x \in \mathcal{D}_\wr} \left(f_0(x) + \sum_{i=1}^{m} a_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x) \right) \tag{2.18}$$

The function $\Lambda(\mathbf{x}, \mathbf{a}, \boldsymbol{\nu})$ is affine and concave with respect to $\mathbf{a}$ and ν. One important property of the dual is that it is a lower bound of $f_0(\mathbf{x})$. It is trivial to show that:

$$g(\mathbf{a}, \boldsymbol{\nu}) = \inf_{x \in \mathcal{D}_\wr} \Lambda(\mathbf{x}, \mathbf{a}, \boldsymbol{\nu}) \leq \Lambda(\mathbf{x}, \mathbf{a}, \boldsymbol{\nu}) \leq f_0(\mathbf{x}). \tag{2.19}$$

Fig. 2.6 shows a toy example with $f_0(x)$ in blue and its dual function in red. Note that the dual function meets the $f_0(x)$ around the center of the graph. This will bring us to the next important concept of duality.

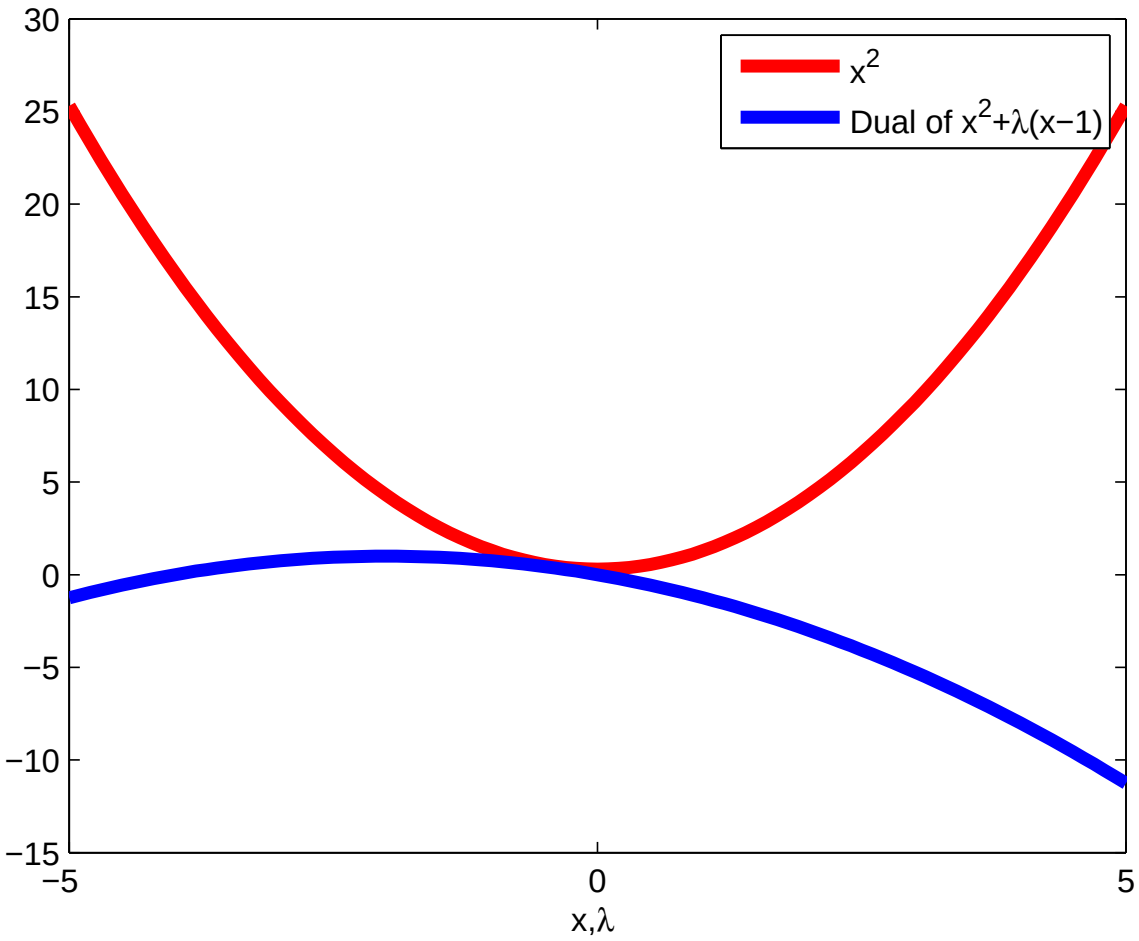


Figure 2.6: A toy example of $f(x)$ and in blue and $g(a)$ in red.

2.2.2 Lagrange Dual Problem

The next step is to find the maximum value of the dual, getting the smallest lower bound of the original problem:

$$\begin{aligned} d^* = \underset{\mathbf{a},\nu}{\text{maximize}} \quad & g(\mathbf{a}, \boldsymbol{\nu}) \\ \text{subject to} \quad & \lambda_i > 0, \quad i = 1, \ldots, m. \end{aligned}$$

.

The values obtained are referred to as the dual optimal $\mathbf{a}, \boldsymbol{\nu}$. Weak duality occurs when $d^* < p^*$ and strong duality occurs when $d^* = p^*$ and usually occurs when $f_0(\mathbf{x})$ and $f_i(\mathbf{x})$ are convex.

2.2.3 Estimation

Maximum Likelihood Estimation

Maximum likelihood estimation [86] is a frequentest method used to estimate the parameters of a parametric distribution. The advantages include good convergence properties as the number of training samples increases. Maximum likelihood estimation can often be simpler than alternate methods, such as Bayesian techniques, but the method is sensitive to outliers. More formally, suppose we have $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, .., \mathbf{x}_N\}$ independent samples drawn where $\mathbf{x} \sim P(\mathbf{x}|\theta)$. Where θ is some set of parameters and $P(\mathbf{x}|\theta)$ is the parametric distribution, because each sample is drawn independently the likelihood with respect to θ can be written in the form:

$$P(\mathbf{X}|\theta) = \prod_{n=1}^{N} P(\mathbf{x}_n|\theta) \tag{2.20}$$

In many cases it is more convenient to work with the log-likelihood function denoted as:

$$\tilde{l}(\theta) = \sum_{n=1}^{N} ln(P(\mathbf{x}_n|\theta)) \tag{2.21}$$

We can then write our solution formally as the argument that θ maximizes the log likelihood 2.21, where the $\hat{\theta}$ denotes an estimate.

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\{\tilde{l}(\theta)\} \tag{2.22}$$

Expected Maximization Algorithm

Expected maximization (EM) [87] algorithm is an iterative approach to solve for models with latent variables using maximum likelihood estimation. There are several ways the EM algorithm can be used to find hidden variables used to model complex distributions. We will explore only the case for mixture models, including Gaussian mixture models and HMM. We define a latent variable z, for example, in a mixture model:

$$P(\mathbf{x}|\theta) = \sum_{k=1}^{K_{em}} p(\mathbf{x}|\theta_k, z = k)p(z = k). \tag{2.23}$$

Where $\theta = \{\theta_1, .., \theta_{K_{em}}\}$ is the set of parameters. It is not difficult to show that the likelihood function for N samples can be written in vector form:

$$P(\mathbf{X}|\theta) = \sum_{\mathbf{z}} p(\mathbf{X}|\theta, \mathbf{z})p(\mathbf{z}). \tag{2.24}$$

Where $\mathbf{z} = [z_1, .., z_N]$ is a vector for every random sample, this means for every $\mathbf{x}_n$ there is a variable z_n that can take on K_{em} and the summation is over K_{em}^N elements. We would like to determine the set of parameters θ without any access to z. Therefore, the EM algorithm is used.

Each iteration of the EM algorithm is composed of two steps: an estimation (E) step and a maximization (M) step. The E-step calculates the expectation over a set of latent variables z. The M-step then maximizes these variables. Consider an iterative step of the EM algorithm to optimize $\tilde{l}(\theta)$; at some step the $l - th$ value is denoted by $\tilde{l}(\theta_l) = ln(P(\mathbf{X}|\theta_l))$. The E-step allows one to calculate a set of functions bounded above by $\tilde{l}(\theta)$ for every value of θ_l. With some clever manipulation, it can be shown that:

$$\tilde{l}(\theta) - \tilde{l}(\theta_l) \geq \sum_{\mathbf{z}} P(\mathbf{z}|\mathbf{X}, \theta) ln \left(\frac{p(\mathbf{X}|\mathbf{z}, \theta)P(\mathbf{z}|\theta))}{(P(\mathbf{z}|\mathbf{X}, \theta_l)P(\mathbf{X}|\theta_m)} \right) \tag{2.25}$$

Letting the right of equation 2.25 equal to $\Delta(\theta|\theta_l)$, where $\Delta(\theta = \theta_l|\theta_l) = 0$, one is now able to create a function bounded above by $\tilde{l}(\theta)$:

$$\tilde{l}(\theta) \geq \tilde{l}(\theta_l) + \Delta(\theta|\theta_l) = \tilde{l}(\theta|\theta_l). \tag{2.26}$$

It can be shown that we can find the value θ that maximized $\tilde{l}(\theta|\theta_l)$:

$$\theta_{l+1} = \underset{\theta}{\operatorname{argmax}}\{\sum_{\mathbf{z}} p(\mathbf{z}|\mathbf{X}, \theta_l) ln(p(\mathbf{X}, \mathbf{z}|\theta)p(\mathbf{z}|\theta))\}. \tag{2.27}$$

It is not difficult to see that 2.27 is an expected value with respect to the posterior probability **z**.

Instead of optimizing $\tilde{l}(\theta)$ the algorithm calculates and optimizes $\tilde{l}(\theta|\theta_l)$. The process as demonstrated in Fig. 2.7, θ_1 is randomly initialized, then the E-step determines $\tilde{l}(\theta|\theta_1)$ in black. Then the value that optimizes $\tilde{l}(\theta|\theta_1)$ is used to calculate $\tilde{l}(\theta|\theta_2)$ and then $\tilde{l}(\theta|\theta_2)$ is maximized to calculate θ_3 until some stopping criteria is met.

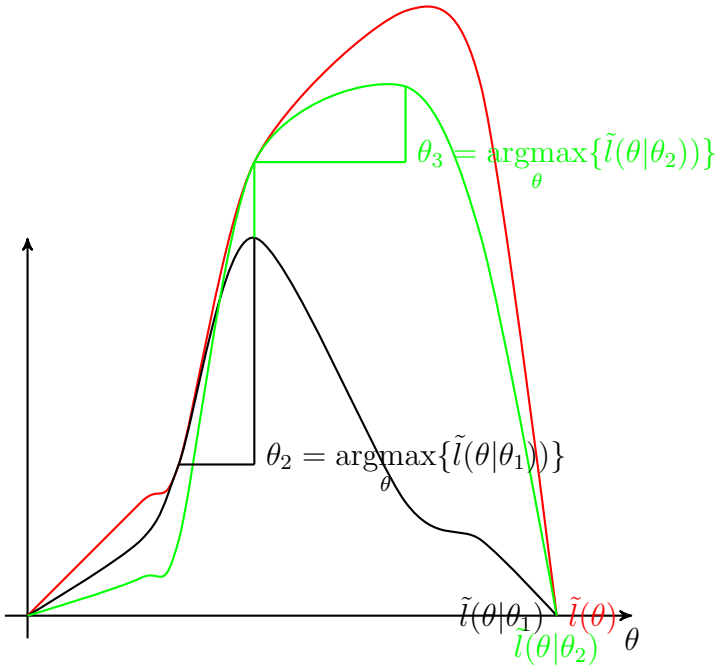


Figure 2.7: Two iterations of the EM, the first iteration is in yellow, and the second iteration is in green.

The function $\tilde{l}(\theta|\theta_l)$ is difficult to calculate but in some cases can be simplified; if the mixture is in the form of 2.23 the optimization problem is simplified to:

$$\tilde{l}(\theta|\theta_l) = \sum_{k=1}^{K_{em}} \sum_{n=1}^{N} p(z = k|\mathbf{x}_n, \theta_l) ln(p(\mathbf{x}_n, z = k|\theta))\} \tag{2.28}$$

Assuming there is a closed-form solution for the maximum for L iterations the complexity is $O(NK_{em}L)$.

An important approximation of mixture models is the k-means clustering, where $p(\mathbf{x}|z = k, \theta) = \mathcal{N}(\mathbf{x}|, \mu_k, \epsilon I)$. It can be shown that as $\epsilon \longrightarrow 0$ we can determine cluster membership by using the EM algorithm to minimize the following:

$$C(\mathbf{z}, U) = \sum_{k=1}^{K_m} \sum_{\mathbf{n=1}}^{N} z_{n,k}(\mathbf{x}_n - \boldsymbol{\mu}_k)^2. \tag{2.29}$$

Where $U = [\boldsymbol{\mu}_1, .., \boldsymbol{\mu}_{K_m}]$ is the cluster centroid and $z_{n,k}$ is a binary variable. Another form of the EM algorithm is the Baum Welch algorithm used for HMM [55]. The idea is essentially the same

but there is a dependence assumption between hidden variables. Using the nomenclature from [55] we replace the hidden variables z with a set of T variables $\mathbf{Q} = \{q_1, .., q_T\}$ representing the hidden state variables. The variable X is replaced with $\mathbf{O} : \{\mathbf{o}_1, .., \mathbf{o}_T\}$ observations. The state transitional probability coefficients $a_{ij} = P(q_t = s_j | q_{t-1} = s_i)$ with an associated transitional probability matrix A and initial state distribution: $\pi_j = P(q_1 = s_j)$, where $\boldsymbol{\pi} = [\pi_1, .., \pi_M]$ and the observation distribution $P(\mathbf{o}_t | q_t = s_j, \hat{\theta}_j) = b_j(\mathbf{o}_t)$ with parameters $\hat{\theta}_j$ determines how likely a $\mathbf{o}_t$ is at a particular time step t. We can estimate the probability of a sequence using the hidden states to formulate a more complex distribution. The likelihood is given by 2.30

$$P(\mathbf{O}, \mathbf{Q}|\theta) = \prod_{t=1}^{T} \prod_{(i,j)}^{K_{HMM}} (b_j(\mathbf{o}_t))^{I[q_t = s_j]} (a_{ij})^{I[q_t = s_j \wedge q_{t-1} = s_i]} (\pi_i)^{I[q_1 = s_i]}. \tag{2.30}$$

The parameters θ can now be estimated iteratively by maximizing:

$$\tilde{l}(\theta, \theta_{l-1}) = \sum_{\mathbf{Q}} P(\mathbf{Q}|\mathbf{O}, \theta_{l-1}) \log(P(\mathbf{O}, \mathbf{Q}|\theta)). \tag{2.31}$$

Other related clustering methods used in this work, but not theoretically explored, include [88] and [89]. A closed form solution is not necessary for the EM algorithm but is extremely desirable as iterative methods are extremely difficult to incorporate into the algorithm [89]. Early work in this thesis used HMM but free parameter selection was difficult for large data sets. Also research on TRECVid and MediaEval used support vector machines.

2.3 Regression, Classification and Kernels

Classification and regression are two supervised machine learning problems that are both used in this thesis. Both classification and regression have many of the same issues such as over-fitting and bias. As these issues are better illustrated with regression, we will spend a little more time exploring them.

2.3.1 Regression

In this section, we will focus on linear regression with Gaussian distributed noise. We will also review quadratic regularization, which also assumes a Gaussian distribution. The reason is there exists a closed-form solution. The properties of these methods have been well studied; they are an artifice for introducing kernels and have comparable performance to other methods.

Linear Basis Function Models (LBFM) are analogous to linear regression, but, instead of the dependent function h being a linear combination of the explanatory variables $\mathbf{x}$, the dependent variable is a linear combination of fixed nonlinear functions $\boldsymbol{\phi}(\mathbf{x})$. The function $\boldsymbol{\phi}(\mathbf{x})$ is a function of the explanatory variables. The value of the parameters $\mathbf{w}$ is to be determined, where $(\mathbf{w})_0 = w_0$ is referred to as the bias parameter, not to be confused with statistical "bias". The relationship can be written in the form:

$$h(\mathbf{x}) = \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}) = w_0 + \sum_{l=1}^{L-1} w_l\phi_l(\mathbf{x}). \tag{2.32}$$

Where $\boldsymbol{\phi}(\mathbf{x}) \in \Re^L$ is a vector-valued function with the first element equal to one, symbolically: $(\boldsymbol{\phi}(\mathbf{x}))_0 = 1$. The vector $\mathbf{w} \in \Re^L$ is the set of parameters. The function $h(\mathbf{x})$ is explicitly a function of $\mathbf{x}$, the variable $\mathbf{w}$ is simply a parameter.

Parameter Estimation

In order to model the parameters we assume that a target Γ can be modelled by a deterministic function $h(\mathbf{x})$ and the difference between the observations and the deterministic function is due to additive random noise ξ, symbolically:

$$\Gamma = \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}) + \xi \tag{2.33}$$

If we assume that the noise ξ is Gaussian distributed, the PDF can be written in the form:

$$P(\Gamma|\boldsymbol{\phi}(\mathbf{x}), \mathbf{w}, \sigma^2) = \mathcal{N}(\Gamma|\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}), \sigma^2) \tag{2.34}$$

If we denote the set of observations with the vector $\boldsymbol{\Gamma} = [\Gamma_1, \Gamma_2, ..., \Gamma_{N-1}, \Gamma_N]^T$ where $\boldsymbol{\Gamma} \in$

$\Re$ and the set of observations $\Phi^T = [\boldsymbol{\phi}(\mathbf{x}_1), \boldsymbol{\phi}(\mathbf{x}_2), ..., \boldsymbol{\phi}(\mathbf{x}_{N-1}), \boldsymbol{\phi}(\mathbf{x}_N)]$, the likelihood can be written in the form:

$$P(\mathbf{\Gamma}|\Phi, \mathbf{w}, \sigma^2) = \prod_{n=1}^{N} \mathcal{N}(\Gamma_n|\boldsymbol{\phi}(\mathbf{x}_n), \mathbf{w}, \sigma^2) = \mathcal{N}(\mathbf{\Gamma}|\Phi\mathbf{w}, \sigma^2 I) \tag{2.35}$$

Taking the log we obtain the log likelihood:

$$ln(P(\mathbf{\Gamma}|\Phi, \mathbf{w}, \sigma^2)) = \sum_{n=1}^{N} ln(\mathcal{N}(\mathbf{\Gamma}_n|\boldsymbol{\phi}(\mathbf{x}_n), \mathbf{w}, \sigma^2))). \tag{2.36}$$

Maximizing equation 2.36 by taking the gradient, one can obtain a closed form solution for the parameters:

$$\widehat{\mathbf{w}} = \sum_{n=1}^{N} (\sum_{n=1}^{N} \boldsymbol{\phi}(\mathbf{x}_m)\boldsymbol{\phi}(\mathbf{x}_m)^T)^{-1}\boldsymbol{\phi}(\mathbf{x}_n)\Gamma_n \tag{2.37}$$

It is sometimes convenient to write out equation 2.37 using the design matrix:

$$\hat{\mathbf{w}} = (\Phi^T\Phi)^{-1}\Phi^T\mathbf{\Gamma} \tag{2.38}$$

The error of the model is given by the difference of the residual between the deterministic function and the training set:

$$\hat{\sigma}^2 = \frac{1}{N}\sum_{n=1}^{N}(\Gamma_n - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n))^2. \tag{2.39}$$

Assuming the noise is Gaussian and $\boldsymbol{\phi}(\cdot)$ is the correct function, the value should converge to the variance of noise, but there are many challenges associated with selecting the proper $\boldsymbol{\phi}(\cdot)$. In the next section we will review the methods using a simple polynomial function; it should be noted that many of the methods here can be used in selecting other parametric and non-parametric methods and also apply to the classification problems in this thesis.

Selecting a Model

A major problem associated with linear basis functions and many other free parameters is selecting a good $\boldsymbol{\phi}(\mathbf{x})$. This section can be viewed as an allegory for the problems encountered in every

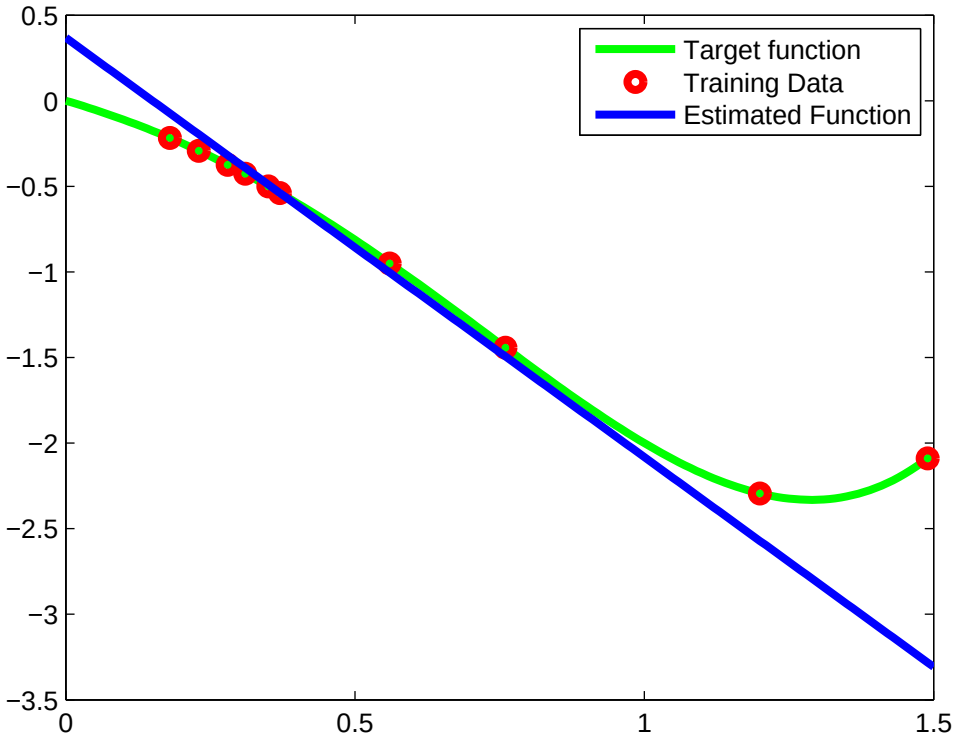


Figure 2.8: Under-fitting example linear function used to model a tenth order polynomial with training samples in red.

chapter from determining the number of states in a model to the best kernel. Model selection can be done empirically by using cross-validation, but as $\mathbf{x}$ gets large or $\phi(\cdot)$ is complex this becomes infeasible. For example, consider using a simple linear function to fit a fifth order polynomial function as shown in Fig. 2.8. It is evident that the line is too constrained to fit all the training points. It seems that building a more complex model will solve the problem, but this will introduce the more interesting problem of over-fitting.

The main problem with complex models is over-fitting. Consider Fig. 2.12, a simple function in green with red samples used for training. The blue function is a tenth order polynomial estimated using 2.38. It is evident that the new function does not accurately represent the target function. This type of error is known as over-fitting, when the model is too complex and models the training data but does not accurately reflect the actual underlying process.

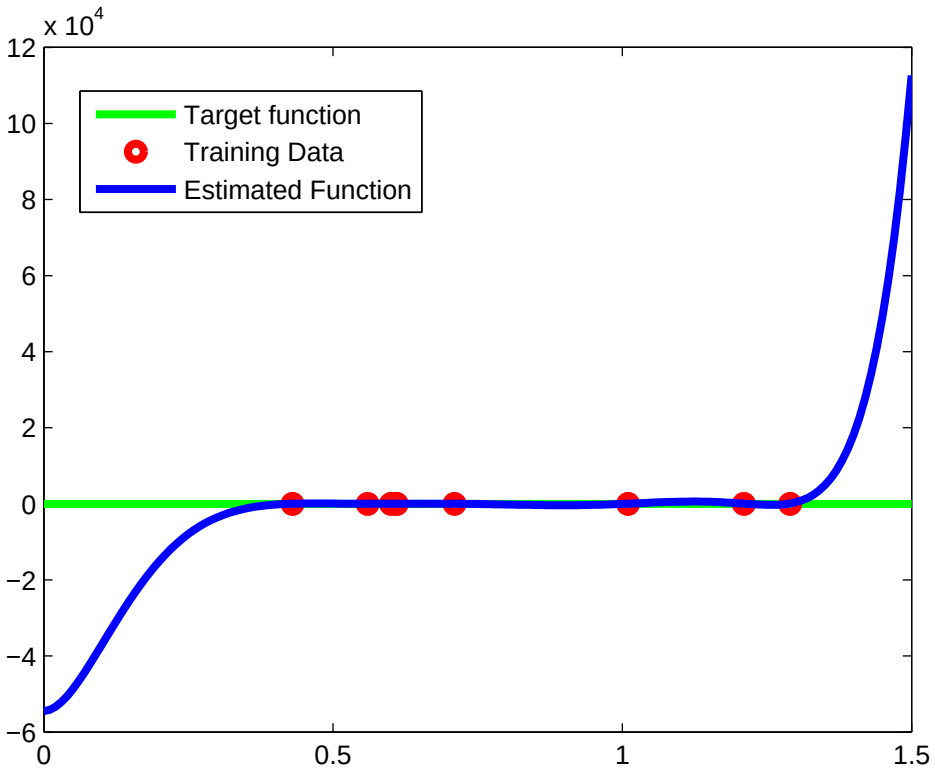


Figure 2.9: Over-fitting example tenth order polynomial used to model a simple function with training samples in red.

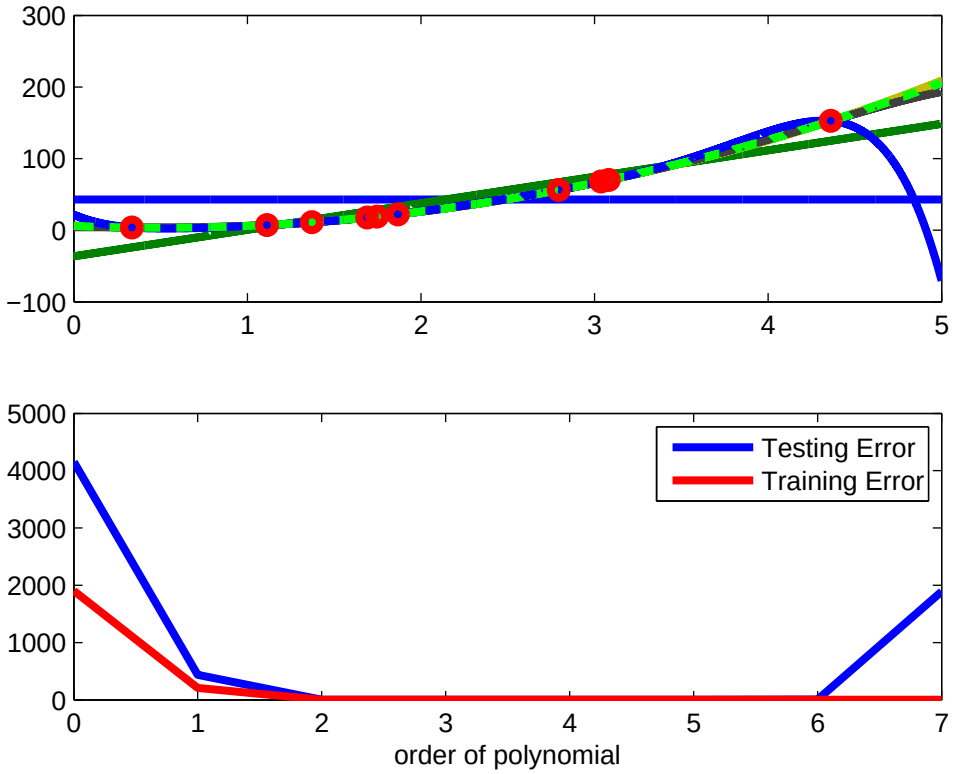


Figure 2.10: Top: Polynomial functions of different orders. Bottom: Training error and validation error of different order models.

The relationship between model complexity and model error is shown in Fig. 2.10. The top plot shows different order polynomials fitting samples generated by a third order polynomial while the bottom plot represents residual squared training error and testing error with respect to different order polynomials. It is evident that the training error of the training data decreases relative to the order of polynomials. In contrast, the error of the testing data is minimal when the estimated polynomial has the same order of training data. Even if the model is correct the over-fitting problem can still occur.

Random noise and lack of training samples also contribute to over-fitting. As well as affecting the residual error, random noise causes over-fitting. The problem is caused by the model fitting the noise. Lack of training samples can also affect the results. The more complex the model, the more training is required to train it. The results of too much noise and lack of data is shown in

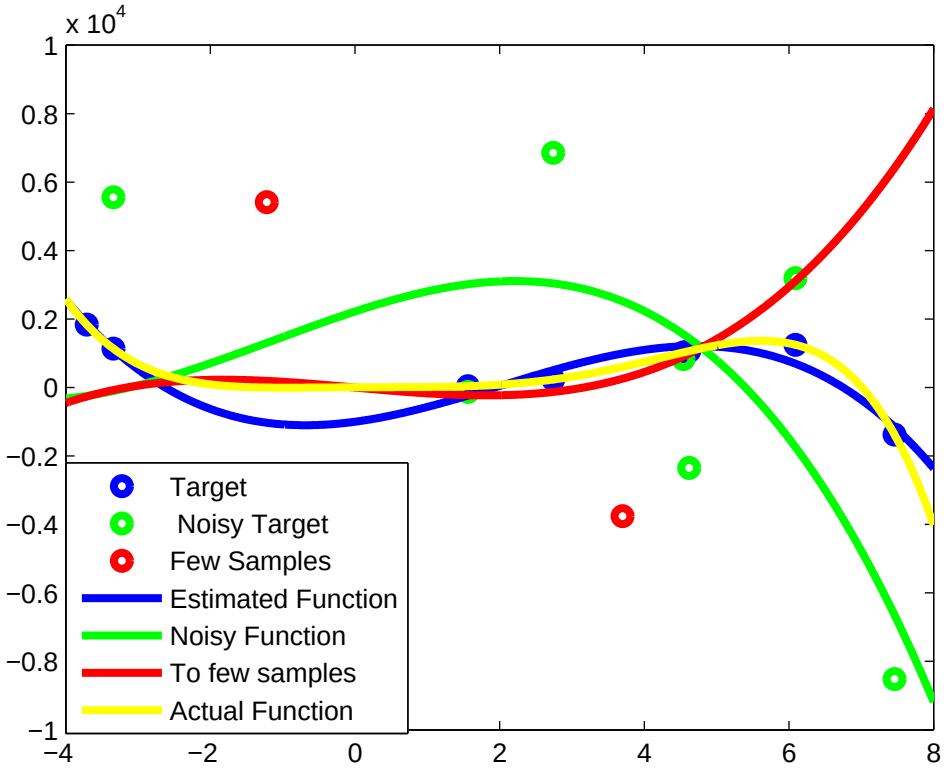


Figure 2.11: Effect of random noise and fewer training samples on estimated data.

Fig. 2.11, with samples represented by the small circles and the associated estimated function with the same color. The actual target function is in yellow. The samples are generated by a sixth order polynomial with all the estimated functions having the same order. The target examples in blue are samples with some noise, the function estimated using these points does well at tracking the target function. The green points represent points that have relatively more noise; the function does poorly at tracking the actual function. Finally, the function in red has many fewer training samples. It is evident that the estimated function tracks the training points, but it is not representative of the actual function. To overcome these problems, we will use regularization and kernels.

Ridge Regression

Ridge regression [90] is a regularized version of linear regression and was introduced in the early 70's. It can be used to ensure numerical stability and the parameters are less prone to over-fitting.

Taking a Bayesian approach, more precisely the MAP estimate, one can introduce a prior distribution over the coefficients $\mathbf{w}$. There are many kinds of distributions, but, for simplicity, we will consider only a zero-mean Gaussian distribution of the form:

$$P(\mathbf{w}|\mathbf{0}, \alpha_r I) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha_p I) = (\frac{\alpha_p}{2\pi})^{\frac{L}{2}} e^{\frac{-\alpha_p}{2}\mathbf{w}^T\mathbf{w}}. \tag{2.40}$$

Using Baye's theorem, the posterior distribution for $\mathbf{w}$ is proportional to the product of the prior distribution and the likelihood function:

$$P(\mathbf{w}|\Phi, \beta I) \propto P(\mathbf{\Gamma}|\Phi, \mathbf{w}, \beta I)P(\mathbf{w}|\mathbf{0}, \alpha_p I), \tag{2.41}$$

where α_p is the precision of the distribution. The zero mean simply implies that we make prior assumptions that the parameters are close to zero. Using Baye's theorem, the posterior distribution for $\mathbf{w}$ is proportional to the product of the prior distribution and the likelihood function. We can maximize equation 2.41 directly but it involves completing the square and several manipulations. Taking the negative logarithm of 2.41 the cost function becomes:

$$\mathcal{L}(\mathbf{w}) = \sum_{n=1}^{N}(\Gamma_n - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n))) + \alpha_p\mathbf{w}^T\mathbf{w} \tag{2.42}$$

This form is simpler to deal with and has an intuitive meaning. Fig. 2.12 demonstrates how the regularization term forces the parameters closer to zero. In blue is the original cost function and in red is the regularization term that has a minimum at zero. The addition of the regularization term to the cost function makes the minimum closer to zero, but the error for the training data is larger.

The solution of equation 2.42 is given by:

$$\mathbf{w}_{MAP} = (\Phi^T\Phi + \alpha_p I)^{-1}\Phi^T\mathbf{\Gamma}. \tag{2.43}$$

To see how regularization helps with numerical stability, let us remember that a matrix with positive eigenvectors is invertible. The matrix $\Phi^T\Phi$ is positive definite; therefore, its eigenvalues λ_i are positive with corresponding eigenvectors $\mathbf{u_i}$. It is not difficult to show that any eigenvalues of $\Phi^T\Phi + \alpha_p I$ is $\lambda_i + \alpha_p$, because α_p is positive, the matrix $\Phi^T\Phi + \alpha_p I$ is also invertible. In addition, if λ_i is numerically close to zero, 0 adding α_p makes the eigenvectors larger. Selecting

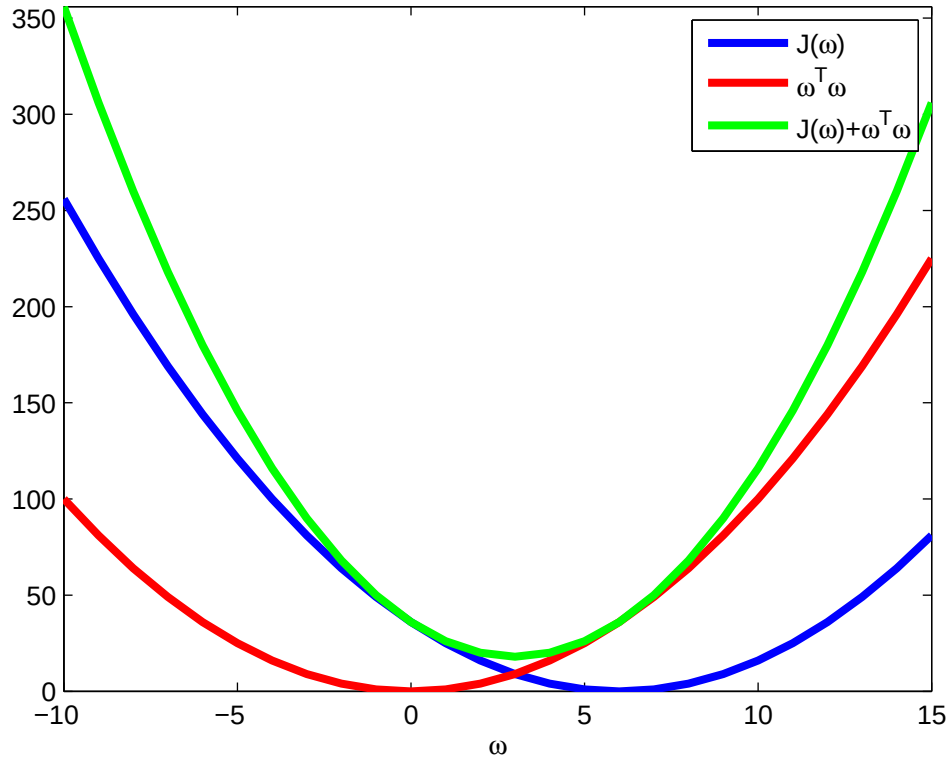


Figure 2.12: Cost function, quadratic regularization and addition of both.

the appropriate value for α_p is determined using cross validation. Ridge regression adds error to the model but decreases variance, two topics that will be discussed in a later section.

Variable Selection

Regularization of $\mathbf{w}$ using the $L1$ norm has sparse solutions: many of the estimated coefficients of $\mathbf{w}$ are zero. When the goal is to reduce the number of features, we can use [91]. If variables are correlated we can use Elastic Net [92] that is a combination of both the $L1$ and $L2$ regularization. As in ridge regression, there is a α_p parameter referred to as the lambda parameter that is selected empirically to determine minimum error on the validation set. If $(\mathbf{w})_l$ is zero, at the minimum error, that feature is not necessary for prediction.

Bias-Variance Trade-off

In this section, we introduce Bias-Variance Tradeoff [93], a fundamental concept in model selection. The method applies to real value functions using the squared loss. We study the method because the concept of bias and variance applies to many learning algorithms and analytically explains many of the above observations. Although it is a general method, we examine it with respect to linear regression for illustrative purposes. Let $f(\mathbf{x})$ be some function we would like to determine for some hypothesis $h(\mathbf{x})$ to approximate $f(\mathbf{x})$.

Given some data set $\mathcal{D}$, the squared loss is given by:

$$\mathcal{L}_{\mathcal{D}}(h) = \mathcal{E}_{\mathbf{x}}((h(\mathbf{x}|\mathcal{D}) - f(\mathbf{x}))^2). \tag{2.44}$$

The quantity is over only one data set; therefore, one would like to know how the error behaves over all possible data sets. Thus, we take the expected value over all possible data sets and decompose the squared error into bias and variance.

$$\mathcal{L}(h) = \mathcal{E}_{\mathcal{D}}(L_{\mathcal{D}}(h)) = \mathcal{E}_{\mathcal{D}}\{\mathcal{E}_{\mathbf{x}}((h(\mathbf{x}|\mathcal{D}) - f(\mathbf{x}))^2)\} \tag{2.45}$$

The average hypothesis is an approximation of the expected hypothesis as is given by:

$$\mathcal{E}_{\mathcal{D}}((h(\mathbf{x}|\mathcal{D}))\} = \bar{h}(\mathbf{x}) \approx \frac{1}{K_f}\sum_{k=1}^{K} h(\mathbf{x}_n|\mathcal{D}_k) \tag{2.46}$$

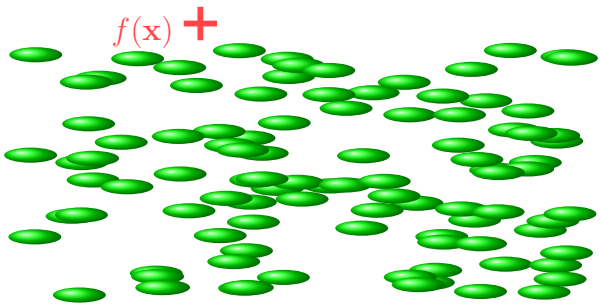


Figure 2.13: An example of a complex model with high variance and low bias where green dots represent possible hypotheses and the red dot represents the target function.

It can be shown with several steps that:

$$\mathcal{L}(h) = \mathcal{E}_{\mathbf{x}}((\bar{h}(\mathbf{x}) - f(\mathbf{x}))^2) + \mathcal{E}_{\mathcal{D}}\{\mathcal{E}_{\mathbf{x}}((h(\mathbf{x}|\mathcal{D}) - \bar{h}(\mathbf{x}))^2)\} = bias(h) + var(h)$$

Complex models usually have low bias and high variance; this is because there are more parameters creating a larger hypothesis space. The bias is related to the amount of error in the model; the variance represents how much the model changes with different data-sets. As a result, complex models can learn more complex functions $f(\mathbf{x})$ but have a high variance as a direct response of this flexibility. Simpler models are much more constrained but in turn have much less variance. This is demonstrated comparing Fig. 2.13, which represents a complex model and Fig. 2.14, which represents a simple model. In both figures the green circles represent possible realizations of some hypothesis $h(\mathbf{x}|\mathcal{D}_k)$ for different data sets $\mathcal{D}_k$. The actual target function $f(\mathbf{x})$ represents in red some hypothetical hypothesis space $\mathcal{H}$. Many of the green circles of the more complex model in Fig. 2.13 come extremely close to $f(\mathbf{x})$ implying low bias, but because there are more parameters there is a more complex hypothesis space. Furthermore many of the values are farther away. The simpler model has a high bias because the model is too simple to fit the function $f(\mathbf{x})$, but because the model is more constrained it has less variance.

If Bias-Variance Trade-off is performed on Γ where $\Gamma = f(\mathbf{x}) + \xi$ we get the following expression:

$$\mathcal{L}_{\mathcal{D}}(h) = \mathrm{E}_{\mathbf{x}}((h(\mathbf{x}|\mathcal{D}) - \Gamma)^2) \tag{2.47}$$

Decomposing as above, we get:

Figure 2.14: An example of a simple model with low variance and high bias in some hypothesis space where the red represents the actual function and the green represents possible hypothesis.

$$\mathcal{L}(h) = \mathrm{E}_{\mathbf{x}}((\bar{h}(\mathbf{x}) - f(\mathbf{x}))^2) + \mathrm{E}_{\mathcal{D}}\{\mathrm{E}_{\mathbf{x}}((h(\mathbf{x}|\mathcal{D}) - \bar{h}(\mathbf{x}))^2)\} + \sigma^2 = bias(h) + var(h) + noise$$

This final expression explains many of the observations mentioned above. The bias term $\bar{h}(\mathbf{x})$ represents the best estimate of data and represents the inherent error of the model; this is sometimes referred to as deterministic noise. Once a threshold has been reached no new data can fix the problem. The variance characterizes the over-fitting problem. This term is dependent on the data, therefore, high variance can be decreased with more data. The noise term cannot be reduced. To control variance we will use regularization and in Chapter 3 we will use some ensemble methods. As we use many kernel methods much of our bias and variance will be controlled by free parameters in the kernel.

Kernel Ridge Regression

Kernel methods became popular with the inception of support vector machines, and we will use this section extensively in Chapter 4. Kernel ridge regression [94] has a close form optimum making it ideal for state based methods in Chapter 4. As demonstrated in the previous section, we can use a basis function $\boldsymbol{\phi}(\mathbf{x})$ to transform a feature space into a non-linear space and still use a linear algorithm. The main problem is $\boldsymbol{\phi}(\mathbf{x})$, which may be difficult or impossible to calculate. Kernels allow one to express functions without calculating $\boldsymbol{\phi}(\mathbf{x})$ directly. First we will define the error term that is the analog to the noise term $\boldsymbol{\xi} = [\xi_1, ., \xi_N]^T$ that can be defined as:

$$\boldsymbol{\xi} = (\boldsymbol{\Gamma} - \Phi\mathbf{w}) \tag{2.48}$$

One can derive the cost function as a minimization problem in vector form as:

$$\begin{aligned} \underset{\xi,\mathbf{w}}{\text{minimize}} \quad & \boldsymbol{\xi}^T\boldsymbol{\xi} + \alpha_p\mathbf{w}^T\mathbf{w} \\ \text{subject to} \quad & (\boldsymbol{\Gamma} - \Phi\mathbf{w}) = \boldsymbol{\xi} \end{aligned}$$

Letting $\mathbf{a}$ be the vector of Lagrange multiplier or dual variables, we can use Lagrange multipliers to convert the constrained problem to an unconstrained problem:

$$\Lambda(\boldsymbol{\xi}, \mathbf{w}, \mathbf{a}) = \boldsymbol{\xi}^T\boldsymbol{\xi} + \alpha_p \mathbf{w}^T\mathbf{w} + \mathbf{a}^T(\boldsymbol{\Gamma} - \Phi\mathbf{w} - \boldsymbol{\xi}). \tag{2.49}$$

Using the KKT conditions to find the minimum value of 2.49 given by:

$$\mathbf{w} = \frac{1}{2\alpha_p}\Phi^T\mathbf{a}. \tag{2.50}$$

This equation states that the parameter vector $\mathbf{w}$ is a linear combination of the columns Φ^T; more specifically it is a linear combination of the training samples $\boldsymbol{\phi}(\mathbf{x_n})$. Substituting back into the original optimization problem the dual function becomes:

$$g(\mathbf{a}, \boldsymbol{\xi}) = \frac{1}{4\alpha_p}\mathbf{a}^T\Phi\Phi^T\mathbf{a} + \boldsymbol{\xi}^T\boldsymbol{\xi} + \mathbf{a}^T\boldsymbol{\Gamma} - \mathbf{a}^T\boldsymbol{\xi} \tag{2.51}$$

Solving with respect to $\mathbf{a}$, we get $\boldsymbol{\xi} = \frac{1}{2}\mathbf{a}$, the importance of the value proportional to noise or residual. Substituting this value of $\mathbf{a}$ into equation 2.52 and defining the gram matrix as $\mathbf{K} = \Phi\Phi^T$, we get:

$$g(\mathbf{a}) = -\frac{1}{4\alpha_p}\mathbf{a}^T\mathbf{K}\mathbf{a} - \frac{1}{4}\mathbf{a}^T\mathbf{a} + \mathbf{a}^T\boldsymbol{\Gamma} \tag{2.52}$$

Minimizing by differentiating with respect to $\mathbf{a}$:

$$\mathbf{a} = 2\alpha_p(\mathbf{K} + \alpha_p\mathbf{I})^{-1}\boldsymbol{\Gamma} \tag{2.53}$$

Inserting 2.50 and 2.58 into 2.32:

$$\begin{aligned} h(\mathbf{x}) =& \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}) = \frac{1}{2\alpha_p}(\Phi^T\mathbf{a})^T\boldsymbol{\phi}(\mathbf{x}) = \frac{1}{2\alpha_p}(\Phi^T(2\alpha_p(\mathbf{K} + \alpha_p\mathbf{I})^{-1}\boldsymbol{\Gamma}))^T\boldsymbol{\phi}(\mathbf{x}) \\ &= (\mathbf{K} + \alpha_p\mathbf{I})^{-1}\boldsymbol{\Gamma})^T\Phi\boldsymbol{\phi}(\mathbf{x}) = (\mathbf{K} + \alpha_p\mathbf{I})^{-1}\boldsymbol{\Gamma})^T\mathbf{k}(\mathbf{x}). \end{aligned}$$

Because $y(\mathbf{x})^T = y(\mathbf{x})$, one can write the final form of the new expression as:

$$h(\mathbf{x}) = \mathbf{k}(\mathbf{x})^T((\mathbf{K} + \alpha_p\mathbf{I})^{-1}\boldsymbol{\Gamma}) \tag{2.54}$$

The main advantage of kernels is $\boldsymbol{\phi}(\mathbf{x})$ does not have to be calculated directly, avoiding the explicit introduction of the feature vector, allowing one to use feature spaces of high, even infinite, dimensionality. The next section demonstrates why in a more general setting.

2.3.2 Kernels

Representer Theorem

As several different kernel methods are used in this thesis, we will demonstrate why they work so well by describing Representor's Theorem [95], a general assertion that states that the hypothesis in the form $h(\mathbf{x}) = \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x})$, can be constructed with training samples in such a way that minimizes:

$$min_{\mathbf{w}} \left\{ \frac{1}{N}\sum_{n=1}^{N} \mathcal{L}(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n), \Gamma_n) + \alpha_r \mathbf{w}^T\mathbf{w} \right\}. \tag{2.55}$$

To better understand the problem, consider the decomposition of $\mathbf{w} = \mathbf{w}_s + \mathbf{w}_\perp$. Where $\mathbf{w}_s \in \mathcal{S}$, such that $\mathcal{S} \in span\left\{\boldsymbol{\phi}(\mathbf{x}_1), \boldsymbol{\phi}(\mathbf{x}_2), ..., \boldsymbol{\phi}(\mathbf{x}_{N-1}), \boldsymbol{\phi}(\mathbf{x}_N)\right\}$, i.e be the component of $\mathbf{w}$ that is spanned by the training examples and $\mathbf{w}_\perp$ be perpendicular to $\mathbf{w}_s$. This is demonstrated in Fig. 2.15 where $\mathbf{w} \in \Re^3$ and $\mathbf{w}_s \in \Re^2$ spanned by the training examples in red.

Examining equation 2.58 it is simple to show that if we use the spanning criteria the loss function does not change:

$$\mathcal{L}(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n), \Gamma_n) = \mathcal{L}(\mathbf{w}_s^T\boldsymbol{\phi}(\mathbf{x}_n), \Gamma_n) \tag{2.56}$$

This is because $\mathbf{w}_\perp^T\boldsymbol{\phi}(\mathbf{x}_n) = 0$, because $\|\mathbf{w}\|^2 \geq \|\mathbf{w}_s\|^2$, then the equation can be solved in the form of kernels:

$$\hat{h}(\mathbf{x}) = \sum_{n=1}^{N} a_n \kappa(\mathbf{x}_n, \mathbf{x}) \tag{2.57}$$

Where $\mathbf{a}$ is the solution to the following minimization problem:

$$min_{\mathbf{w}} \left\{ \frac{1}{N}\sum_{n=1}^{N} \mathcal{L}(\sum_{m=1}^{N} a_n \kappa(\mathbf{x}_n, \mathbf{x}_m), \Gamma_n) + \alpha_r \sum_{(n,m)=1}^{N} a_n a_n \kappa(\mathbf{x}_n, \mathbf{x}_m) \right\}. \tag{2.58}$$

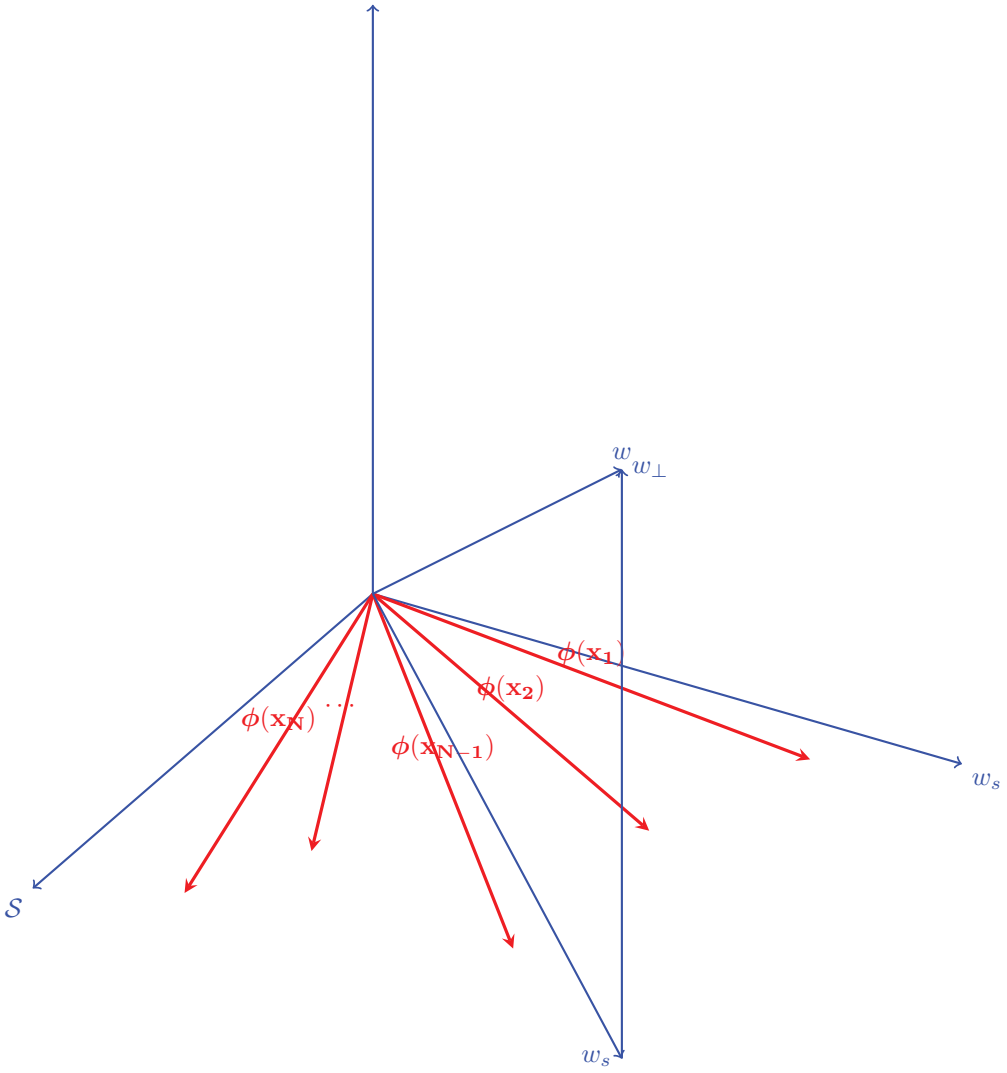


Figure 2.15: $\mathbf{w} \in \Re^3$ and $\mathbf{w}_s \in \Re^2$ spanned by the training examples in red.

$$\phi(\mathbf{x}) = [x_L^2, .., x_1^2, \sqrt{2}x_L x_{L-1}, .., \sqrt{2}x_L x_1, \sqrt{2}x_{L-1}x_{L-2}, .., \sqrt{2}x_{L-1}x_1, .., \sqrt{2}x_2 x_1, \sqrt{2c}x_L, .., \sqrt{2c}x_1, c]^T \tag{2.60}$$

There is a large amount of work on kernel methods; this work will select kernels by minimizing generalization error or in the case of clustering some predefined criteria.

Kernels also allow one to work in spaces of much higher dimensions, the polynomial basis functions grow exponentially, yet the kernel for a polynomial basis function of order d is given by:

$$\kappa(\mathbf{x}, \mathbf{z}) = \boldsymbol{\phi}(\mathbf{x})^T \boldsymbol{\phi}(\mathbf{z}) = (\mathbf{x}^T \mathbf{z} + c)^d \tag{2.59}$$

The feature vector for $d = 2$ is given in equation 2.60, which is extremely large. It is not difficult to verify that working with the kernel involves $\mathcal{O}(L)$ computation, but to calculate $\boldsymbol{\phi}(\mathbf{x})$ explicitly rewires $\mathcal{O}(L^d)$.

Another kernel that will be used is the Radial basis function kernel (RBF kernel). The RBF kernel is excellent for dealing with data with nonlinear properties, but is sensitive to outliers. The RBF kernel is given by:

$$\kappa(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\sigma_{RBF}^2}\right) \tag{2.61}$$

and can be viewed as a dot product between two non-countable vectors:

$$\exp\left(-\frac{1}{2}||\mathbf{x} - \mathbf{x}'||^2\right) = \sum_{j=0}^{\infty} \frac{(\mathbf{x}^\top \mathbf{x}')^j}{j!} \exp\left(-\frac{1}{2}||\mathbf{x}||^2\right) \exp\left(-\frac{1}{2}||\mathbf{x}'||^2\right). \tag{2.62}$$

In order to determine if $\phi(\cdot)$ is a valid kernel $\mathbf{K}$ must be positive semi-definite for a finite sequence [96].

Sparse Kernel Machines

The Vapnik Chervonenkis Dimension (VC-Dimension) is an important concept with respect to classification and illustrates important concepts in machine learning. It can be shown that the difference between the test error and training error is given by:

$$P\left(\text{Test Error} \leq \text{Training Error} + \sqrt{\frac{\mathbf{VC}(\log(2N/\mathbf{VC}) + 1) - \log(\eta/4)}{N}}\right) = 1 - \eta \tag{2.63}$$

Where N is the number of samples and $\mathbf{VC}$ is the VC dimension, which is usually proportional to the number of parameters, but not always.

It is evident that as $\mathbf{VC}$ gets larger the training error diverges from the testing error and the exact opposite happens as N gets larger. In practice this is not a tight bound, but shows that the learning process is possible and illustrates why training and testing data must be kept separate. It turns out that the loss function for support vector machines has a small VC dimension relative to other methods [83]. There are several other motivations for the Support Vector Machine (SVM), the first with respect to kernel methods is that they only require a subset for training. Unlike the perception learning algorithm, the decision boundary is not dependent on the order in which the data is presented or the initialization of the parameters. Also kernels allow one to deal with non-finite dimensional vector spaces. In the next section we will give a brief overview of missing factors such as a non-separable case, the multi-class problem, soft margin and the relationship to computational learning theory.

2.3.3 Sparse Kernel Machines

Consider the two-class classification problem using linear models shown in equation 2.64. Where $\phi(\mathbf{x})$ is the fixed feature space, transform, with parameters $\mathbf{w}$ and bias parameter b, the input training data $X = \{\mathbf{x}_1, .., \mathbf{x}_N\}$ and target values $\Gamma_1, .., \Gamma_N$ such that $\Gamma_n \in \{-1, 1\}$. Some unknown data point will be classified according to $h(\mathbf{x})$.

$$h(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x}) + b \tag{2.64}$$

We will assume for now that the data set is linearly separable such that there exists a $\mathbf{w}$ and b where if $\Gamma_n = +1$ then $h(\mathbf{x}_n) > 0$ and when $\Gamma_n = -1$ then $h(\mathbf{x}_n) < 0$ for all training data points (for non-linear separable cases, we will use soft margin SVM). It is not difficult to show, given the linearly separable assumption, that $\Gamma_n h(\mathbf{x}_n) > 0$ for all values of n. The decision boundary is a hyperplane. Thus, the perpendicular distance is the only distance of consequence. Therefore, at some point $\phi(\mathbf{x_n})$ the perpendicular distance from the hyperplane is given by:

$$\frac{\Gamma_n y(\mathbf{x}_n)}{\|\mathbf{w}\|} = \frac{\Gamma_n(\mathbf{w}^T\phi(\mathbf{x_n}) + b)}{\|\mathbf{w}\|} \tag{2.65}$$

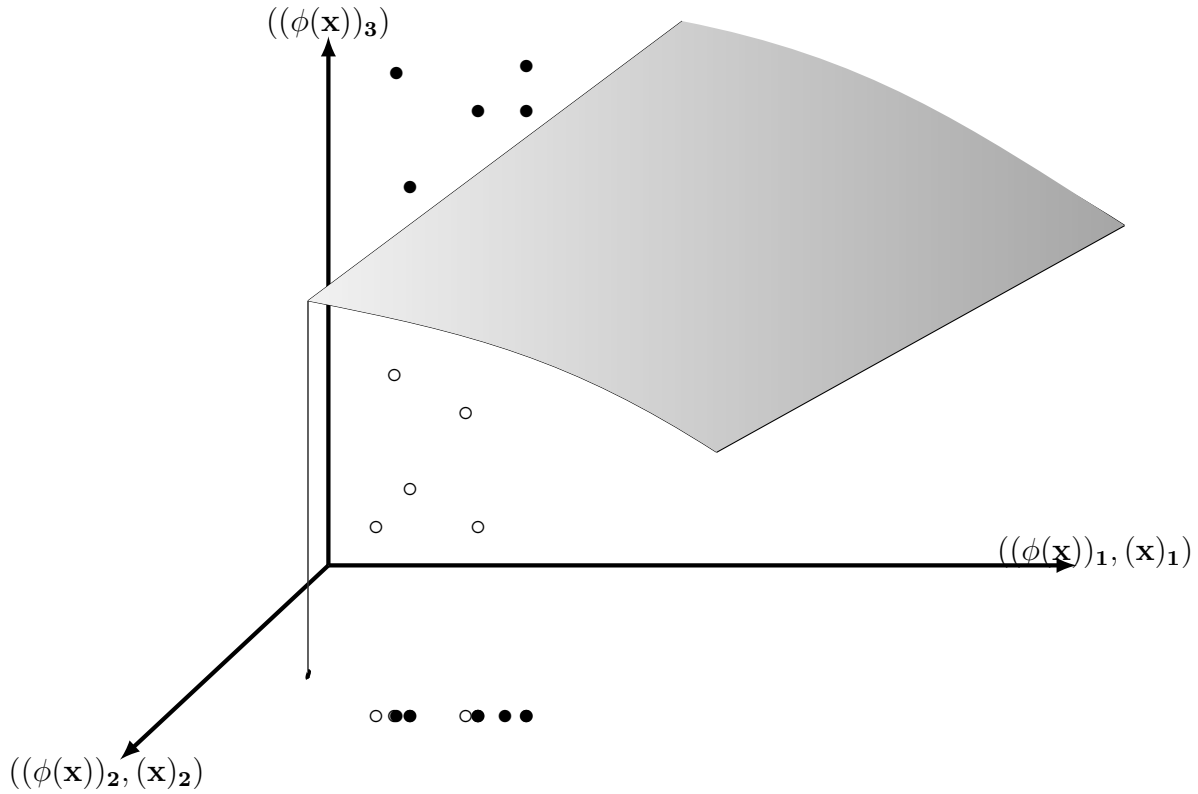


Figure 2.16: An example of using a basis function to map data that is not linearly separable in x to linearly separable dimension. The separating hyper plane is indicated in gray.

The term $\|\mathbf{w}\|$ is used to convert the parameter vector into a unit vector. Using the kernel allows one to map the points into a higher dimensional space, where the data is more likely linearly separable. An example of some data points that are not separable in space $\mathbf{x} \in \mathcal{X}$ but separable in the space $\phi(\mathbf{x}) \in \mathbf{\Phi}$ is shown in Fig. 2.16.

2.3.4 Margin

An important concept in determining the decision boundaries in SVM is the margin ς. The margin is the smallest distance between the decision boundary and all the data points. The expression for the data point that gives the margin is provided in equation 2.66 with an example of the closest data point shown in Fig. 2.17. The training goal of SVM is to solve the parameters $\mathbf{w}$ and b that

maximize the margin. Thus the maximum margin solution is found by solving:

$$\varsigma = \min_n \{\mathbf{w}^T \phi(\mathbf{x_n}) + b\}. \tag{2.66}$$

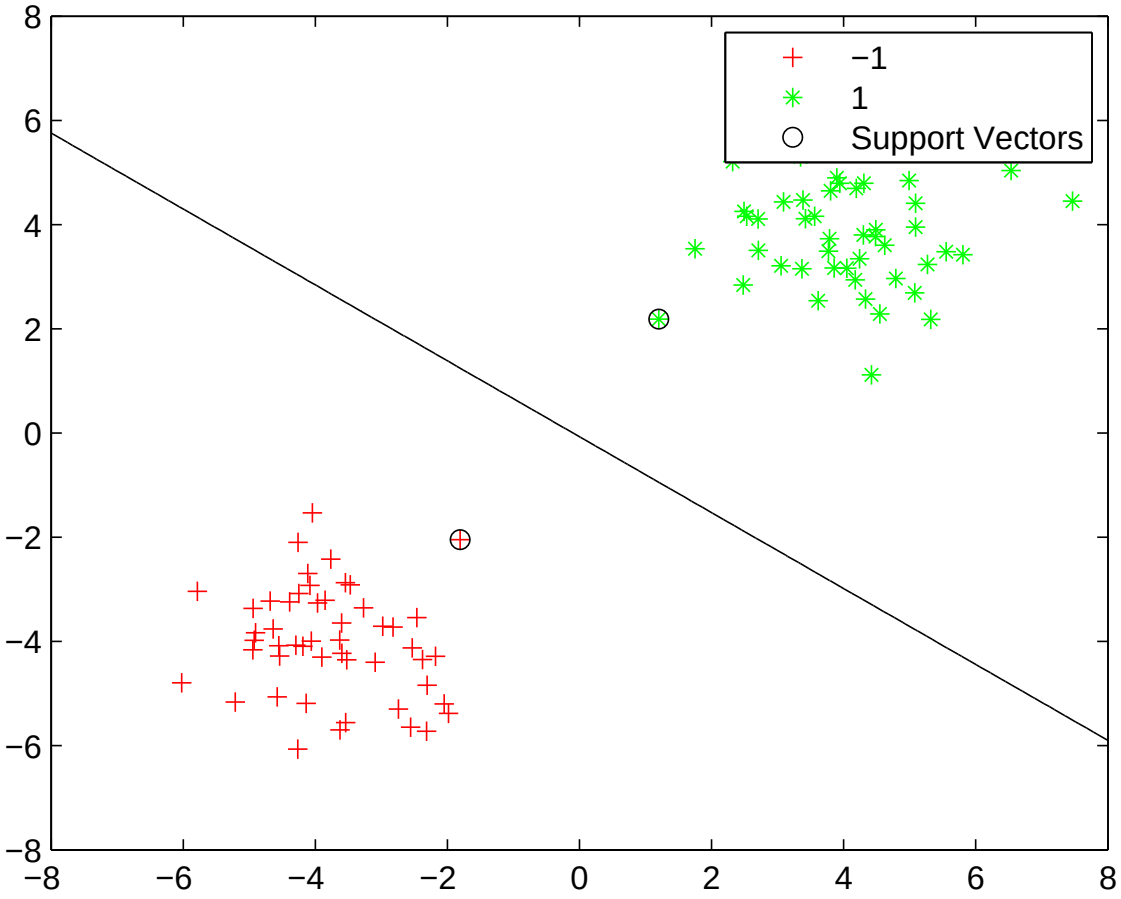


Figure 2.17: Data points with the smallest distance between the decision boundary and any of the samples generated in Matlab.

The training goal of SVM is to solve the parameters $\mathbf{w}$ and b that maximize the margin. Thus the maximum margin solution is found by solving:

$$\max_{\mathbf{w},b} \{\frac{1}{\|\mathbf{w}\|} \min_n \{\mathbf{w}^T \phi(\mathbf{x_n}) + b\}\} \tag{2.67}$$

Although the margin is determined by the training data, more precisely the support vectors, it

can be shown that the margin can be maximized by minimizing the following:

$$\max\left(\mathbf{w}^T\mathbf{w}\right) \tag{2.68}$$

subject to the the following constraints:

$$\Gamma_n(\mathbf{w}^T\phi(\mathbf{x_n}) + b) \geq 1. \tag{2.69}$$

This is known as the canonical representation of the decision hyperplane. In the case of data points for which the equality holds, the constraints are said to be active, whereas for the remainder they are said to be inactive. By definition, there will always be at least one active constraint, because there will always be a closest point, and once the margin has been maximized there will be at least two active constraints. Instead of minimizing $\|\mathbf{w}\|^{-1}$, we maximize the reciprocal with the constraints of 2.70. Using the method of Lagrange multipliers or KKT to incorporate the constraints, the cost function becomes:

$$\mathcal{L}(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{n=1}^{N} a_n\{\mathbf{w}^T\phi(\mathbf{x_n}) + b - 1\} \tag{2.70}$$

Where $\mathbf{a} = [a_1, ..., a_N]^T$ is the vector of Lagrange multipliers. In order for the gradient of the cost function to be a constrained local minimum, it must point in the opposite direction of the feasible region, therefore $a_n \geq 0$. Taking the gradient of $\mathcal{L}(\mathbf{w}, b, \mathbf{a})$ and solving for $\mathbf{w}$ and b and then writing the expression in terms of $\mathbf{a}$ we arrive at the dual representation of the maximum margin problem in which we maximize:

$$\tilde{\mathcal{L}}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m \Gamma_n \Gamma_m \kappa(\mathbf{x}_n, \mathbf{x}_m) \tag{2.71}$$

Where:

$$\sum_{n=1}^{N} a_n \Gamma_n = 0 \tag{2.72}$$

Maximizing equation 2.71; using coordinate ascent is difficult due to the constraints imposed in equation 2.72, therefore, John Platt's sequential minimal optimization algorithm is used [97] .

2.3.5 Kernel K-means and Spectral Clustering

It turns out that k-means can also be expressed with kernels, but, instead of using an iterative algorithm a solution is approximated by relaxing some of the constraints [98]. We follow the tutorial in [99] as it is more compact. To clean up some of the notation but keep it consistent with the previous sections, we define $\hat{\Phi} = \Phi^T$. The k-means cost function can be expressed in matrix form:

$$C = tr((\hat{\Phi} - \bar{M})^T(\hat{\Phi} - \bar{M}))), \tag{2.73}$$

where $\bar{M} = \hat{\Phi}Z\Pi Z^T$ where Π is a diagonal matrix with diagonal elements equal to the number of clusters such that the $k-th$ diagonal element such that:$(\Pi)_{k,k} = (N_k)^{-1}$. The matrix $Z \in \mathfrak{F}_2^{NxKc}$ is an assignment matrix, with elements $(Z)_{n,k} = z_{n,k}$. Each column of $\bar{M}$ corresponds to a centroid, so if the n sample belongs to the $k-th$ centroid the $n-th$ column μ_k. With several manipulations it can be shown that 2.73 can be minimized by maximizing:

$$\begin{aligned} p^* = \underset{\Pi,Z}{\text{maximize}} \quad & \Pi^{\frac{1}{2}} Z^T K Z \Pi^{\frac{1}{2}} \\ \text{subject to} \quad & Z \text{ is binary clustering matrix.} \end{aligned}$$

.

This problem like the SVM problem is difficult to solve, so the constraints are relaxed by introducing the matrix $H = Z\Pi^{\frac{1}{2}}$. It is evident that for a given row of H all the columns are zero except for the column that pertains to having membership. This is demonstrated in Fig. 2.18 where the second sample belongs to the second cluster, i.e., $z_{22} = 1$. We see that in the second row of H all the elements are zeros. It can be shown that $H^T H = I$; as a result, the problem can now be relaxed. The new problem is given by:

$$\begin{aligned} \hat{p}^* = \underset{H}{\text{maximize}} \quad & H^T K H \\ \text{subject to} \quad & H^T H = I. \end{aligned}$$

.

Decomposing $K = U\Lambda U^T$ it can be shown that setting H equal to the K_c Eigenvector with the largest values maximizes the expression i.e $[\mathbf{h}_1, .., \mathbf{h}_{Kc}] = [\mathbf{u}_1, .., \mathbf{u}_{Kc}]$ similar to principle component analysis. One can now perform clustering on H to get cluster membership.

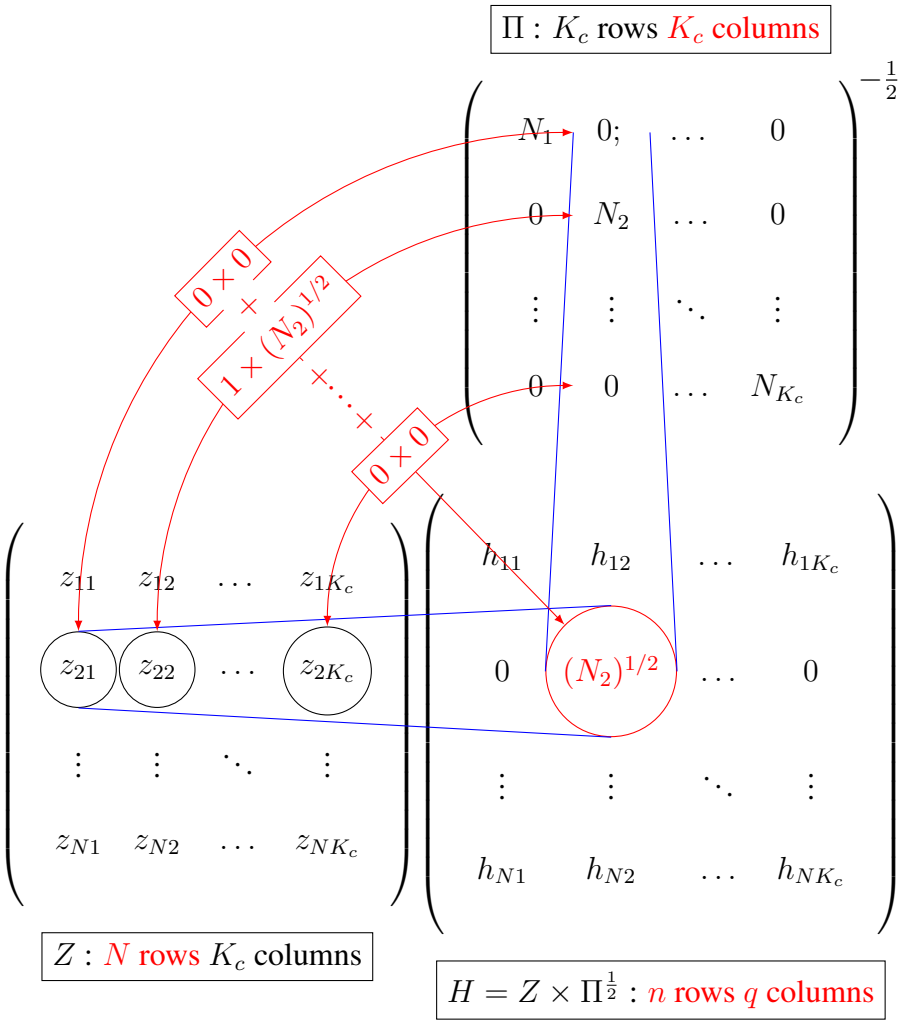


Figure 2.18: Illustration of the relaxation procedure for spectral clustering, the second row of the assignment matrix has all zero elements expect of the 2nd column corresponding to the samples cluster membership. A similar relationship but the nonzero element is proportional to the number of elements in a cluster.

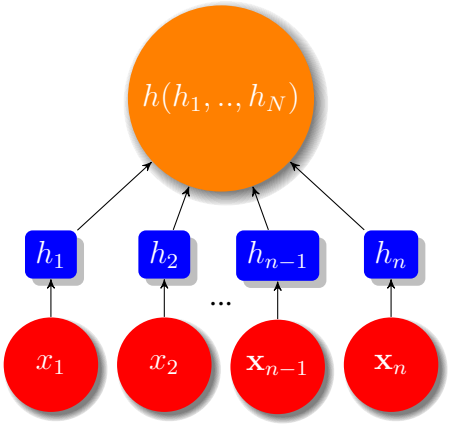


Figure 2.19: Decision-level fusion: An example of Decision-level fusion.

2.4 Multimodal Fusion

In this section, we study two aspects of how to create features and decisions. A central aspect of multimedia processing is the coherent integration of media from different sources and modalities. In addition to the curse of dimensionality, processing of multimedia data requires integration of different modalities. To overcome this problem multimodal fusion methods have been developed. They can be categorized into three types: feature level or early fusion, decision level or late fusion and hybrid multimodal fusion. This thesis will focus on decision-level fusion.

2.4.1 Decision-Level Fusion

Decision-level fusion combines multiple decisions to arrive at a final decision. The strategy has many advantages and decisions are less susceptible to different representations of data. Decision-level fusion allows one to use the most suitable methods for analyzing data; for example, SVM vs hidden Markov models. The drawbacks are loss of decision-level correlation among modalities and the learning process for each level becomes tedious. An example of decision-level fusion is shown in Fig. 2.19; the red nodes represent features, the blue nodes represent outputs of different functions and the function h combines them.

Chapter 3

Decision Fusion Methods for Cognitive Classification of Children's Video Content

This chapter develops several methods to classify children's videos based on their cognitive content. The first section introduces positive developmental video classification for children, a method which automatically classifies videos according to an expertly assigned predefined positive or negative cognitive impact category. The second section develops an automatic age recommendation system for children's video content, such that it recommends the appropriate age for a video using many features including rate of speech. The contributions of this chapter include novel research on cognitive classification of children's video content and a set of features that improve results over state-of-the-art methods when applied to the cognitive classification task.

3.1 Positive Developmental Video Classification For Children

In the realm of child psychology, the impact of media on children is different relative to its impact on adults. In addition to being affected by semantic content, children are also affected by visual and audio sensory bombardment, thereby producing a series of orienting responses that interfere with cognition. Conversely, age and experience allow adults to pay greater attention to informative features, such as dialogue and narrative content [25].

In this section, we define the concept of Positive Developmental Video Classification (PDVC) for children to determine whether a video can be classified according to an expertly assigned predefined positive or negative cognitive impact category. We then use a clustering algorithm to

determine the membership of each frame.

Content with negative effects include: violence, numerous scene changes (referred to as fast-paced in the literature) [18], and sensory bombardment. The effects of negative content include: attention problems, poor social behavior, poor verbal skills, and academic problems [100, 22, 18, 101]. Content that is good for children includes: individuals interacting, socializing, learning, and problem solving [22, 102, 103]. Despite the importance of PDVC, there has been, to our knowledge, no research undertaken in this area with respect to the multimedia and computer vision literature.

Video genre classification appears to be the most obvious solution as education content, such as documentaries, is good for children [103]. But there is lots of content that is good for children that is not classified as educational [100, 22, 18, 101]. In addition there is no one genre that is bad for children, also some content that is bad for children is labeled as educational.

In this work, we develop features based on the observations of the cited psychological literature. Social interaction is an important factor in video developmental effect. This section develops novel cognitive features to determine the amount of social interaction by using novel algorithms based on face detection. These features are more robust than counting the number of faces as used in [1, 47].

Fast-paced programming is bad for children [18]. Research by [18] finds that content with lots of shot boundaries has a negative impact on children's inhibitory control. The use of location of shots is not a conducive global classification feature. As a result, we have developed the shot effect feature.

Audio plays an important role in determining how content is perceived, and audio features have been used in video-content analysis [36, 1, 47, 2, 66]. We extend the work by introducing features used in music information retrieval to capture characteristics of both negative and positive content. We also include affective features due to their link to cognition [2].

Model validation techniques such as K-Fold Cross Validation are commonly used in evaluating the performance of video genre classification (VGC) algorithms [1, 47]. These methods randomly partition the data into training and validation sets. Partitioning the data randomly may lead to the

same series in the training and validation sets. This may cause correct classification due to overfitting because the method may detect similarities between series. To address this concern, we introduce a model validation technique that segments the data with respect to series.

To our knowledge, the contributions of this section include the first work in PDVC for children, the development of novel features, novel decision fusion methods, and the first application of many audio content analysis features to video content analysis. In addition, we introduce a novel database of videos that have been classified as good or bad based on the literature; both TRECVID nor MediaEval has such a database. The method is tested with a model validation technique that prevents classification due to similarities between series. The classification uses majority voting and linear kernel SVM. For smaller feature subsets, the RBF kernel has been used, as free parameter selection was feasible and sensitivity to outliers could be determined. It was found to outperform state-of-the-art video genre classification systems and methods, such as those regarding arousal time curve developed by [2].

3.1.1 Problem Formulation: PDVC

To determine if a video has a positive or negative impact on development, we pose the problem as a two-class problem. We classify each component of the feature vector for every frame of the video as having a positive or negative impact on development. The decision of each classifier is considered a vote. The class of an entire sequence is the class with the most votes. More formally, let j be the label of each class set ω_j, such that ω_0 is the set of negative videos and ω_1 is the set of positive videos. Let $x_{i,t}$ represent the $i-th$ feature and the $t-th$ frame. Let $h_i(x_{i,t})$ represent the decision of the $i-th$ classifier where if $h_i(x_{i,t}) = j$ then $x_{i,t} \in \omega_j$. Based on the majority voting [104, 1, 47] we formulate the decision rule for an entire sequence such that:

$$j = \underset{j}{\operatorname{argmax}}\{\sum_{t=1}^{T}\sum_{i \in S_f} I[h_i(x_{i,t}) = j]\} \tag{3.1}$$

S_f is the set of features. When comparing methods S_f will be a subset of features. The $I[\cdot]$ is the indicator function, which is one, when the condition in the argument is true. Otherwise, it equals zero. We also experiment with the ways in which different sampling rates change the accuracy.

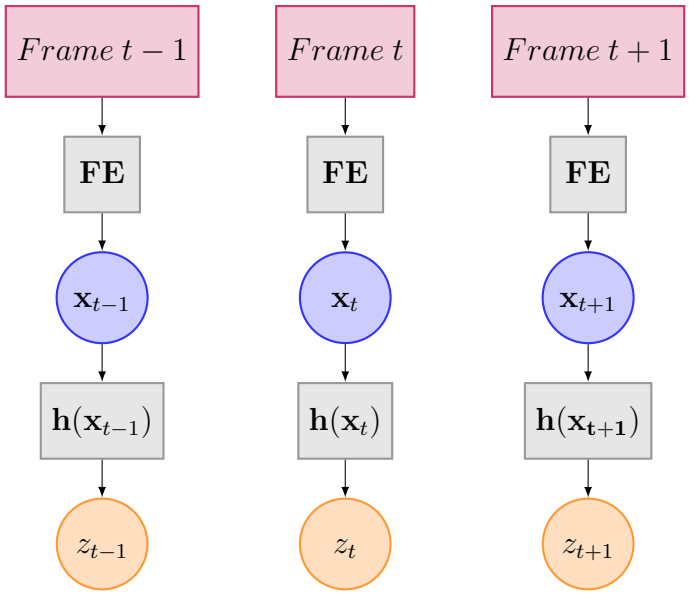


Figure 3.1: Block diagram of the cluster process. Each frame of video goes through feature extraction units (FE). Then the results of the decision units are concatenated into one vector $\mathbf{h}(\mathbf{x_{t+1}})$ and finally the values are passed into the categorical clustering algorithm.

3.1.2 Categorical Clustering: Clustering

To determine the membership of the $t-th$ frame, we use a categorical clustering algorithm. Let $\mathbf{h}(\mathbf{x}_t)$ be a vector containing the decision units of the $t-th$ frame.

$$\mathbf{h}(\mathbf{x}_t) = [h_1(x_{1,t}), .., h_{|S_f|}(x_{|S_f|,t})]^T, \tag{3.2}$$

where $|.|$ indicates the carnality of the set.

Let C_k be a categorical cluster centroid, the $k-th$ cluster centroid with K_{CL} clusters. Let z_t denote the cluster membership of frame t and be obtained using the method in [88], symbolically:

$$z_t = \underset{k}{\operatorname{argmin}} \left(d(C_k, \mathbf{h}(\mathbf{x}_t)) \right), \tag{3.3}$$

where $d(\cdot)$ is the Hamming distance [88]. When $K_{CL} = 2$, the cluster membership of each frame corresponds to a positive or negative impact category. The process is summarized in Fig. 3.1. The results of the decision units are concatenated into a vector and clustering is used.

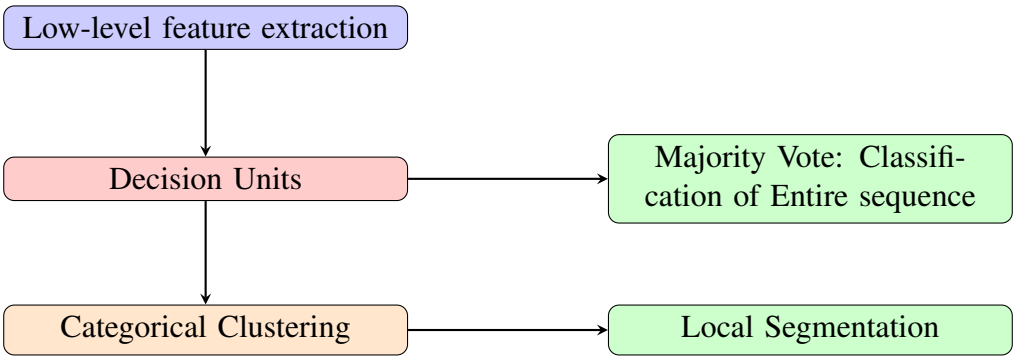


Figure 3.2: The final block diagram combining sequence classification and clustering segmentation.

3.1.3 Block Diagram

The final block diagram is shown in Fig. 3.2. The top block represents the feature extraction block. These features are then used by the decision units. To classify the entire sequence, majority voting of all the decision units is used as depicted by the green block on the left side. Then the outputs of the decision units are placed in a categorical clustering algorithm. The different cluster membership correspond to events that have a positive or negative impact on cognition.

3.1.4 Feature Space:PDVC

Novel Cognitive Features

The novel features developed here use decision-level fusion. The algorithm uses the number of pixels classified as belonging to a face to measure the amount of storytelling, conversation, instruction, and group discussions that are all apparent in content that is good for children [22, 102, 103].

The Face Area Ratio

To capture close-ups, we have developed the face area ratio feature. This is the total number of pixels that a face contain, over the total number of pixels that comprise a video frame. Let S_t be the set of pixels that comprise a $t-th$ frame and let $F_{n,t}$ be the set of pixels that comprise the $n-th$ face such that $F_{i,t} \subseteq S_t$. The face area ratio is given by:

$$FAR = \frac{1}{|S_t|} \sum_{n}^{N_f} |F_{n,t}|. \tag{3.4}$$

The term $|\cdot|$ indicates the cardinality of a set and N_f are the number of faces. The actual face is recognized using the well-known Viola-Jones Face Detection algorithm (VJFD) [105].

Multiple Face Central Moment

To capture the position of the faces in the frame, we define the multiple face central moment as:

$$\bar{x}_t = \frac{1}{N}\sum_{n=1}^{N}\check{x}_{n,t} + \frac{\check{x}_{w,n,t}}{2} \tag{3.5}$$

$$\bar{y}_t = \frac{1}{N}\sum_{n=1}^{N}\check{y}_{n,t} + \frac{\check{y}_{h,n,t}}{2}, \tag{3.6}$$

where $\bar{x}_t$ and $\bar{y}_t$ are central moments of the faces, $(\check{x}_{n,t}, \check{y}_{n,t})$ are the bottom left points of the bounding boxes containing the $n-th$ detected face at frame t and $(\check{x}_{w,n,t}, \check{y}_{h,n,t})$ are the dimensions of the box. The term calculates the mean centre location of all the faces. A face centered in the screen is usually indicative of some kind of verbal interaction with the viewer. Motion features are also included as they have an impact on cognition.

Shot Effect

Content deemed as bad for children incorporates lots of shots. However, shot boundaries are not good for classification using majority voting, because, even with content that includes a relatively large amount of shot boundaries, very few of the overall frames make up shot boundaries. To solve this problem, we introduce the novel structural feature of shot effect. The method models the effect of a shot that seems to decrease with time [18], as an individual with a decaying function. Multiple shots are modeled with a convolution operation with a train of Dirac delta function located at each shot boundary. If the shots are too close together, the function does not have time to decay and the video has a large shot effect. The shot effect of the $t-th$ frame is given by:

$$F_t = \sum_{\tau=0}^{M_s} e^{-\check{\lambda}(t-T_s(\tau))}u(t-T_s(\tau)) \tag{3.7}$$

Were $u(t-T_s(\tau))$ is the well-known step function, $T_s(\tau)$ is the frame index of the $\tau-th$ shot, M_s is the number of shots and λ is the parameter of the exponential function and is determined empirically to maximize classification accuracy.

Spectral Features

Audio plays a crucial role in human perception and determining how content is processed. Audio content that is bad for children has sharp tones to capture a child's attention. This type of audio is associated with a peaky spectrum. To measure tonality of the signal, Spectral Flux [106] is used. Content that is good for children is much more relaxing, which is associated with a flat spectrum. As such, features that measure the flatness of the spectrum are used including: Spectral Crest, Spectral Decrease, Spectral Slope and Spectral Flatness [106, 75]. Dialogue is good for children, therefore the Spectral Rolloff is used to measure the amount of voiced speech [78]. Additionally, we include Spectral Moments such as Spectral Spread, Spectral Skewness, and Spectral Kurtosis [75]. Tonality, harmony, and melody play an important role in characterizing the audio content. The relationship that exists between octave intervals is not always captured by audio models such as Mel Frequency Cepstral Coefficients (MFCC). Therefore, to better describe the audio, we use Pitch Chroma (PC) [107]. The PC is a 12-dimensional Chroma feature that encodes the short-time energy distribution of the underlying music signals over the twelve Chroma bands. To our knowledge this is the first time that such an audio model has been applied to video content analysis. Due to motions link to cognition we include it in under conative features for experimental results.

Table 3.1: Classical features organized by modality with reference. If there is more than one dimension, then $(\cdot)$ indicates dimensionality of feature if larger than one.

Modality	Feature type
Affective	Sound Energy [66], Light Key [66] ,Rhythm [2], Colour Variance [66] and Arousal [2]
Motion	Motion vector energy using motion estimation [36], Average amount of pixels differencing [36], histogram difference
Color	Max color for each channel (3) [36],color histogram moments (9) [36], HSV Histogram (24) [1]
Texture	Wavelet texture variance (12) [1],gray level co-occurrence: Contrast, Correlation, Energy, Homogeneity [1]
Audio	spectral energy [36], spectral centroid [36] ,Pitch [36],Mel Frequency Cepstral Coefficients (8)
Cognitive	Face detection, number of faces[1]

Classical Features

In addition to the classical video classification features [2, 66, 36, 1, 36], we include features from affective analysis due to their link to cognition [2]. All the additional features used are summarized

in Table 3.1, given the link between pace and cognition motion features were tested as cognitive features. As many of the videos were of lower resolution, we use only one block and, as some of the content was black and white the Autocorrelogram was not used.

Table 3.2: Summary of databases: Series, Category, Videos, Genre and Reference with a total of 107 "good" videos and 160 "bad".

Series	Category	Videos	Reference	Genre
Batman	negative	19	[22, 57, 58]	Animation
Baby Einstein(BE)	negative	53	[101]	Live Action
Superman (SM)	negative	22	[22, 57, 58]	Animation
Teletubbies (TT)	negative	11	[22, 108]	Live Action
Brainy Baby (BB)	negative	20	[101]	Live Action
Saturday Morning TV (SMTV)	negative	35	[22, 109]	Live Action
Mister Roger's Neighborhood (MRNH)	positive	23	[22, 102, 103]	Live Action
Dora the Explorer (DE)	positive	14	[22, 102]	Animation
Blues Clues (BC)	positive	13	[22, 102]	Live Action
Sesame Street (SS)	positive	24	[22, 102, 103]	Live Action
Get Along Gang (GG)	positive	21	[103]	Animation
Caillou	positive	12	[18]	Animation

Originally video genre classification features were used for classification as educational content was found to be good for children [22, 102, 103]. As the correlation between fast pace and content was suggested more motion features were incorporated [18]. Finally as the impact of arousal was suggested by [18] affective features were used.

3.1.5 Deterministic Variable Size K-fold Cross Validation

To ensure classification is based on developmental effect and not commonality with the same series, we have developed a different model validation technique. This will be referred to as: Deterministic Variable Size K-Fold Cross Validation (DVSK-Cross Validation). To train the system, the data is partitioned into two sub-sets. The first sub-set consists of all the episodes of one series and is used for validation data. The remaining sub-sets are used as training data and consist of the

remaining series. More formally let $\mathbf{x}_t \in \Re^d$ be the feature vector extracted from the $t-th$ frame such that $(\mathbf{x}_t)_{i,t} = x_{i,t}$. Let $X_l = [\mathbf{x}_1, \mathbf{x}_2, .., \mathbf{x}_T]$ be the feature matrix of the $l-th$ episode of a series. Let χ_m represent the feature matrix of $m-th$ series, containing L_e episodes. One series is comprised of all the episode matrices X_l concatenated together such that:

$$\chi_m = [X_1||X_2||, ..., ||X_{L-1}||X_L]. \tag{3.8}$$

For a given validation set γ_v containing the series v, the training set is given by:

$$\gamma_{train} = \bigcup_{m \neq v} \chi_m. \tag{3.9}$$

This procedure is repeated for every series for a total of twelve times. This methodology has been used to ensure that classification is not based on similarities within the series. Once that data is partitioned, a linear kernel SVM is trained for each element in the feature space and majority voting is used to classify the video. The method can be summarized as follows:

1. concatenate the L_e episode matrix X_l into a series matrix χ_m
2. leave the $v-th$ series out of your training set γ_{train}
3. use γ_v as your validation data
4. repeat until every series has been used in your validation set

An instantiation of DVSK-Cross Validation is shown in Fig. 3.3. The top row contains 20 circles representing different video sequences colored according to each series or fold. The subsequent lines show the training set in red and validation set in blue. For example, the second row shows the validation set in blue corresponding to the first series in the top row colored yellow, and the training set in red corresponds to the series not in the training set. The process is repeated for every fold until each series is used once.

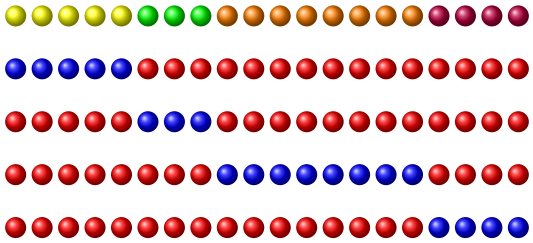


Figure 3.3: An instantiation of DVSK-Cross Validation (K=5) the top row contains 20 circles representing different video sequences colored according to each series or fold, the remaining rows show each iteration with blue corresponding to test data and red corresponding to training data.

3.1.6 Database Summary: PDVC

This chapter classifies the videos as having a positive or negative impact. The videos have been obtained from YouTube and are of varying length even within the same series. The videos showcase different eras, languages, resolutions and aspect ratios. A total of 107 positive impact videos and 160 negative impact videos are used. The video series, as well as expertly assigned predefined positive or negative cognitive impact category, (simply referred to as positive or negative), the number of videos from each series, and the references are all summarized in the first four columns in Table 3.2.

3.1.7 Experimental Procedure: PDVC

Feature extraction has been performed separately; after the features have been extracted they are stored and used for classification. Given the good results of decision fusion and the desire to study categorical clustering feature fusion was not used. Training a decision unit for each feature component and then taking a majority vote. In many cases, individual decision units have good performance and when majority vote is performed on each feature for each frame, the results improve as predicted [110]. It is observed that generally linear kernels perform better as expected for such a large amount of training data. RBF kernels work well for some features but are sensitive to outliers. This was determined in the initial experiments using affective features, where performance was sensitive to the RBF scaling factor. As a result, linear kernels were used instead.

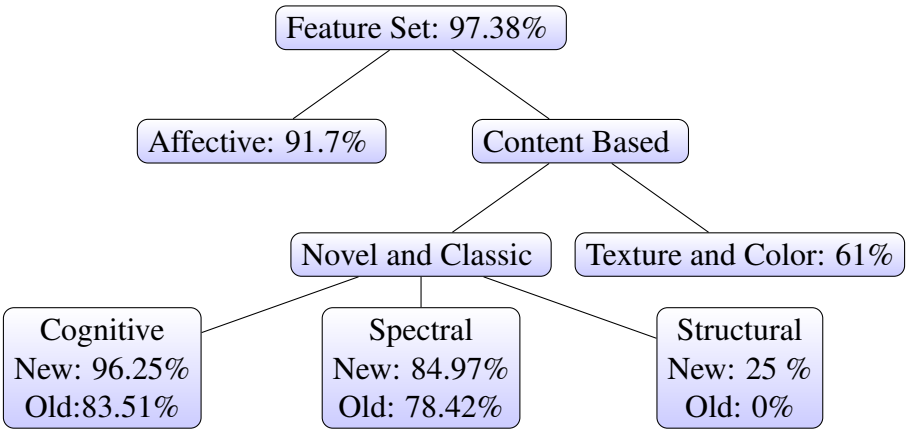


Figure 3.4: Tree summarizing experimental results.

Table 3.3: Confusion matrix: All Features in bold compared to method used in [1], [2]

Class	Predicted Bad	Predicted Good
Actual Bad	(**155**,92,29)	(**5**,68,131)
Actual Good	(**2**,18,54)	(**105**,89,53)

3.1.8 Experimental Results: PDVC

Fig. 3.4 provides a high level summary of the performance accuracy of the novel features and the features listed in Table. 3.1, organized by modality. The top nodes demonstrate the results of using all the features, with 97.38% accuracy. It is evident that the modalities that are the children of the node *Novel and Classic* had better performance than the texture and color features. The better performance suggests a correlation between the cognitive content and the modalities, hence novel features were developed in these modalities. By combining the novel content based features with the classic content based features, we see an improvement as shown in Fig. 3.4.

As stated above, the algorithm using all the features has an accuracy of 97.38% compared to [1], with an overall accuracy of 67.79%. Using the ATC developed in [2] as a feature, results in 31.00% accuracy. The confusion matrix is shown in Table 3.8, each column of the confusion matrix represents the instances in a predicted class, while each row represents the instances in an actual class (the method developed here is in bold) followed by the method in [1] and [2], respectively.

It is enlightening to examine the performance over series, as shown in Table 3.4. The accuracy of the novel method is in column six and is compared to the method used in [1] and displayed in column seven. Violent TV series such as Batman and Superman have 100% accuracy. Non-violent videos such as Baby Einstein and Brainy Baby are more difficult to classify as bad, but they still have over 90% accuracy. Saturday morning TV has the worst results of all the bad series, stemming from the larger variety in content. In the good series category, Sesame Street has the worst accuracy. One possible explanation is that this series has instances of fast-paced scenes used to generate excitement that may be confused with violence. Comparing the results used in [1], it is evident that many of the misclassification results using this method involve animated videos, which is not surprising given that the method is designed for video genre classification that has many color and texture features that had an accuracy of 61%.

Table 3.4: Summary of database and classification accuracy of the novel method compared to the state of the art in VGC [1] and ATC used in [2]. The columns represents: video series, category, accuracy of novel method, VGC and ATC , respectively where *A* corresponds to animation and *L* corresponds to live action.

Series	Category	Novel Accuracy	VGC Accuracy	ATC Accuracy
Batman (*A*)	negative	100%	100%	10.5%
Baby Einstein (*L*)	negative	98.83%	96.11%	0%
Superman (*A*)	negative	100%	100%	13.63%
Teletubbies (*L*)	negative	100%	100%	91.67%
Brainy Baby (*L*)	negative	95%	0%	68.18%
Saturday Morning TV (*A/L*)	negative	91.43%	0%	0%
Mister Roger's Neighborhood (*L*)	positive	100%	100%	66.67%
Dora the Explorer (*A*)	positive	100%	78.75%	100%
Blues Clues (*L*)	positive	100%	100%	0%
Sesame Street (*L*)	positive	95.83%	100%	70.83%
Get Along Gang (*A*)	positive	100%	0%	0%
Caillou (*A*)	positive	100%	0%	58.33%

The precision and recall of the three methods is shown in Table 3.5, where a true positive is a video with a positive impact. We see that the system has vastly superior precision and much better recall than [1] and [2].

Table 3.5: Precision and Recall.

Bad Series	Precision	Recall
Novel Method	0.95	0.98
Method [1]	0.56	0.83
Method [2]	0.28	0.49

Table 3.6 demonstrates the accuracy of the novel features organized by modality using Equation 3.11; the set of features that comprise the modality are used for S_f. We see that the novel cognitive features, spectral features, and audio model perform extremely well. This suggests that audio plays an important role in how content impacts children. It was found there was little music in the content that was good for children, this may be correlation not causation. The location and size of the face also exhibits good accuracy because content that is good for children has a large number of characters interacting, which is often accompanied by close-ups of the face. The above features all have accuracy of over 80%, individually outperforming the method in [1] that has an accuracy of 67.79% and [2] that has an accuracy of 31.00%. The novel structural feature of shot effect has improved accuracy over location of shots but has an accuracy of only 25%. This suggests a strong correlation between the novel features and the expertly assigned predefined impact categories. This is not surprising because the features have been developed and selected based on the observation of the experts who assigned the predefined impact categories.

It is evident that the new features outperform the old features of the same modality. Further examining the first row of Table 3.6, we compare the number of faces to our novel features based on face detection. The novel method has 12.75% better classification accuracy, and it should be noted that in this data set there are typically no crowds of people that may be confused with storytelling. The second row shows the novel spectral features compared to all the audio features in Table 4.1. The new features have over 6.5% better performance. The third row compares the Pitch Chroma to the MFCC [36]. This feature has almost 20% better classification. Finally the Shot Effect is compared to shot location and has 25% better performance, but other similar affective features have better performance, which will be explored in other sections.

Table 3.6: Average accuracy of novel features compared to state-of-the-art features organized by modality.

Modality	Video Content Analysis	Novel Features
Cognitive	83.51%	96.25%
Spectral	78.42%	84.97%
Structural	0%	25%

Analysis of Novel Features

The novel features appear less susceptible to the genre, examining Table 3.4 we see most of the misclassification are with respect to animation. This is most likely that the texture and color features detected attributes associated with animation, none of the novel features used color or texture. Examining Table 3.4, we see that the ATC misclassified relatively fast paced shows that are good for children like Sesame Street and Blues Clues, the novel features did not seem as susceptible to motion that may occur in some content that is good for children as did the ATC. One important observation is the excellent performance of affective features that will be expanded on in the next section.

Affective Features

Affective features also have good performance, with the RBF kernels having slightly better performance than linear kernels. The method using all the arousal features has 245 correct classifications out of 267 with a 91.7% accuracy. Comparing arousal features to valence features, we see that valence has an accuracy of 85.50% which is smaller than arousal. Valence does better in many cases but has 0% accuracy when classifying Dora the Explorer, Blues Clues and Teletubbies. Therefore, this is difficult to determine and more study is needed. Using the ATC developed in [2] results in 31% accuracy. This poor performance result is most likely due to the comparability criteria. The normalization and scaling requirements imposed likely make it difficult when comparing videos, and this is because all videos will have values within the same range.

Table 3.7: Classification accuracy of feature vs series. Each column represents a series while every row represents a feature, where Sound Energy (SE) and Pixel Difference (PD).

Class	Batman	BE	SM	TT	BB	SMTV
Motion	86.21%	100%	100%	75%	72.23%	90.91%
PD	26.31%	64.15%	36.36%	58.33%	31.81%	45.45%
Rhythm	100%	**100%**	100%	91.66%	100%	72.72%
SE	42.11%	**0%**	95.45%	91.67%	100%	96.97%
Class	MRNH	DE	BC	SS	GAG	Caillou
Motion	30%	69.23%	100%	100%	100%	83.33%
PD	63.33%	30.77%	35.71%	41.66%	20.00%	83.33%
Rhythm	100%	100%	21.43%	100%	20%	**100%**
SE	73.33%	100%	92.86%	100%	66.67%	91.67%

The results for each individual feature with respect to each series is shown in Table 3.7. The rhythm component performs extremely well in most cases and performs perfectly on the series Caillou, performing better than Shot Effect. These results agree with the results in [18], where it is suggested that the low rate of scene changes may play a role in why this series has a positive impact on cognition. A similar result is apparent for Baby Einstein. The authors of [101] suggest that the extensive number of scene changes may have a negative impact on cognition. This result is verified by the ability of the rhythm component to obtain perfect classification on this particular series. An interesting observation is that the motion component performs much better than pixel-wise differences. This is different than the results produced in automatic video classification literature which finds little difference [36]. One plausible explanation is that the motion component is less susceptible to changes in illumination. These changes in illumination may increase the amount of motion but have little impact on arousal. Sound energy usually performs well except for Baby Einstein, which has zero percent classification accuracy. This is not surprising because Baby Einstein is designed for younger children. This observation suggests that the target age of the content may play a role in how the arousal features impact cognition. The confusion matrix is shown in Table 3.8. Each column of the confusion matrix represents the instances in a predicted class, while each row represents the instances in an actual class (the method developed here is in bold). The second index represents the method used in [1], the third represents the method used in

Table 3.8: Confusion matrix: Arousal Features in bold compared to method used in [1] ,[2]and valence features.

Class	Predicted Negative	Predicted Positive
Actual Negative	(**150**,92,29,149)	(**10**,68,131,11)
Actual Positive	(**12**,18,54,27)	(**95**,89,53,80)

[2], and the fourth is uses valence features[66, 65].

Fig. 3.5 and Fig. 3.6 demonstrate how the visual arousal features are different for the predefined positive or negative cognitive impact categories. Fig. 3.5 shows the values of pixel differencing motion and rhythm components of two series plotted for successive frames in time. In red are the features generated by the Batman series, a video classified as having a negative impact on cognition, and in blue are the features generated by Mister Rogers' Neighbourhood, a series classified as having a positive impact. It is evident that the magnitude of the arousal features in most of the frames are much larger for Batman.

Fig. 3.6 shows the actual frames from the Batman series (top) and from the Mister Rogers' Neighbourhood series (bottom) that generate the arousal features demonstrated in Fig. 3.5. The frames with Batman contain sudden changes: The first scene begins with Batman and Robin in an aggressive stance. The next scene is Superman fighting a robot, followed by a scene of an explosion. Each of these frames contains rapid changes and flashes, thereby producing arousal manifested in the arousal features (in red) in Fig. 3.5. Compare these to the frames from Mister Rogers' Neighbourhood, in which Mr. Rogers is feeding and observing fish in a fish tank. These are calm scenes and one can see that they produce very different arousal features (in blue) in Fig. 3.5.

Fig. 3.5 shows the lighting key of a video from each series plotted over frames. The negative videos are in red and the positive videos are in blue, and the labels of each series are near the maximum values. It is evident that negative videos have a much larger magnitude. Examining the labels of the negative video, the magnitude of the lighting key does not seem to be related to the genre. For example, Batman has one of the lowest values while Superman has one of the highest. Examining Fig. 3.8 displays the colour variance, and it is observed that the positive video's magnitude on average is much larger than the negative videos. This is not dependent on genre, as

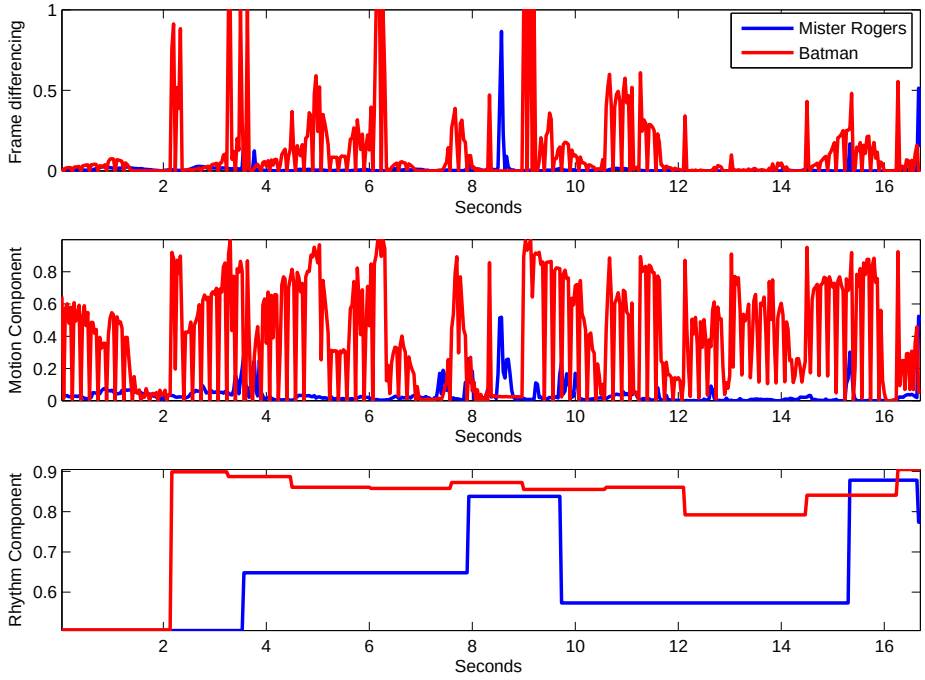


Figure 3.5: Pixel differencing, motion component and rhythm component of two series are plotted for successive frames. In red is the Batman series with negative cognitive impacts and in blue is the Mr. Rogers' Neighborhood series with a positive impact category.

for example the cartoon Caillou has one of the largest magnitudes for color variance while Dora the Explorer has one of the lowest values for colour variance.

Finely adjusting the intervals between each frame used or the sampling rate did not have a major impact on accuracy. For example down-sampling to 1 frame per second did not change the accuracy if all the features where used.

3.2 Results: Clustering

Using categorical clustering on the decision units appears to create segmented periods that have a strong relationship with the expert recommendations. The method seems to work better with videos in the negative impact category as many videos in the positive impact category strictly contain only positive content. Many of the videos in the negative impact category however have a

Figure 3.6: Highly arousing scene from the Batman series (above) and a calming scene from the Mr. Roger's Neighborhood series (below) sampled from the frames used to generate arousal features in Fig. 3.5.

mixture of violent, fast-paced scenes and scary scenes intermingled with scenes that are considered positive, such as scenes showing people interacting, and some educational content.

Examining Fig. 3.9, we see six clips from a Baby Einstein episode. The top of Fig. 3.9 corresponds to videos in cluster one and the bottom corresponds to videos in cluster two. Segments in cluster one consist of clips teaching children to read by individuals reading out text on the screen. These clusters contain the visual presentation and narration that enhance attention [25, 26], and also contain reading that was found to be beneficial [102]. Interestingly these frames contain text but no text detection was used.

The bottom of Fig. 3.9 contains clips from cluster two. All have fast transitions and several clips contain individuals playing loud music or puppets accompanied by strange noises. This content

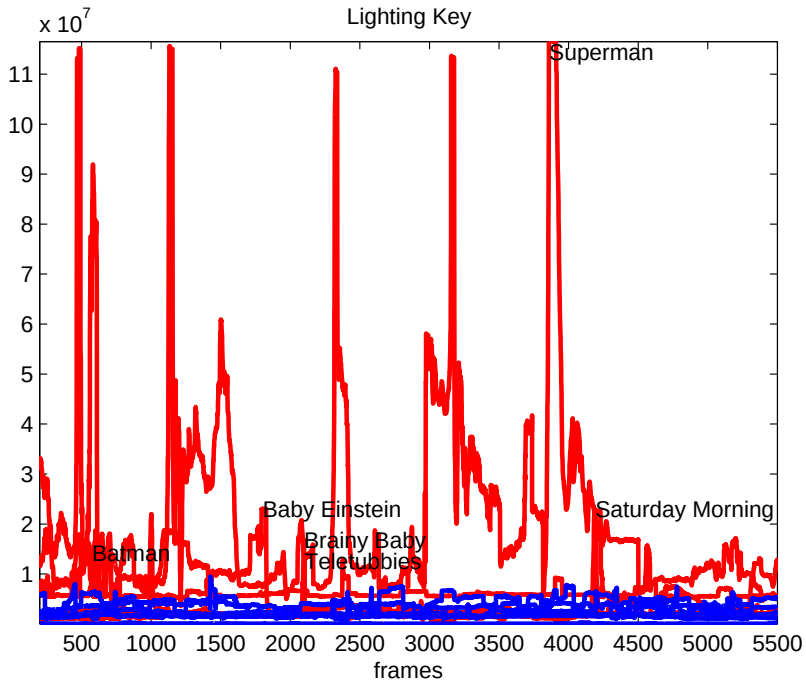


Figure 3.7: Lighting key of videos from each series bad videos in red and blue videos in blue, negative videos only ladled for clarity.

fits the description of stimulating content that has been found to be bad for children [22, 23, 24].

Examining Fig. 3.11, we see the similarities between different clusters from two different series. Examining the top of Fig. 3.11, we see that the clip from Batman consists of a large violent explosion and the clip from Brainy Baby consists of a shot transition between a child's toy and an animated spinning shape. Both are violent and attention grabbing scenes that have a negative impact [22, 57, 18, 58]. Although the violent and attention grabbing scenes were segmented, the algorithm is not explicitly trained to classify or segment these factors. Examining the bottom of Fig. 3.11 we see text similar to that in Fig. 3.9, as previously stated no text detection was included.

The method does not seem to detect the violent scene but rather the stimulating content associated with the violent scene. This is demonstrated in Fig. 3.11. Consider Fig. 3.11 A). We see Superman lifting a large boulder, a fast-paced scene corresponding to cluster two as shown in Fig. 3.11 D). As Superman throws the boulder, the camera tracks the boulder. As shown in Fig. 3.11 B), there is little visual or audio input and the cluster membership changes as shown in Fig. 3.11

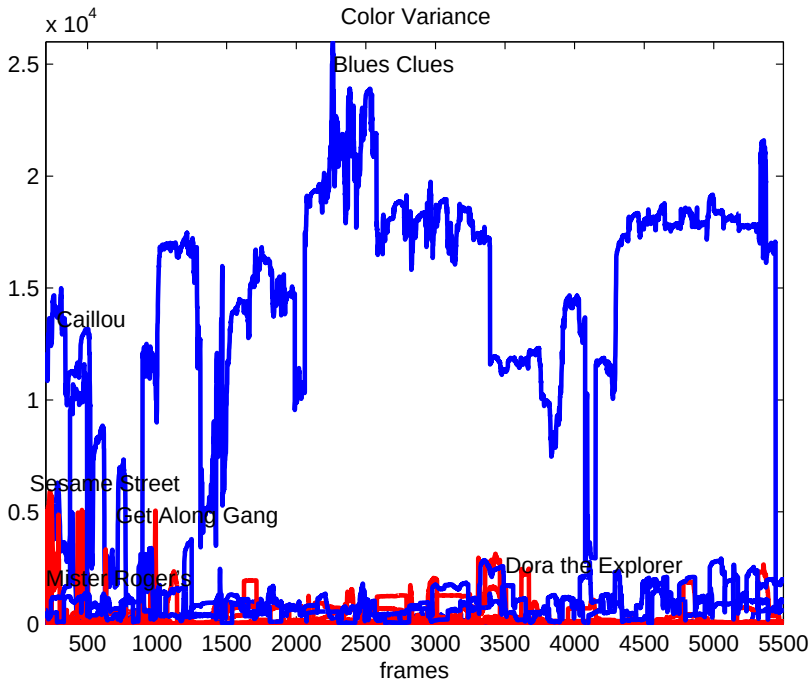
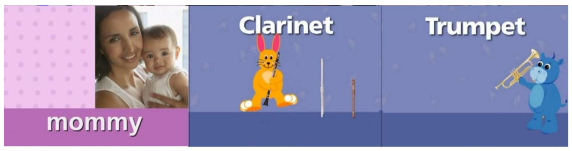


Figure 3.8: Color variance of videos from each series bad videos in red and blue videos in blue, only positive videos only ladled for clarity.

D). In Fig. 3.11 C), we see Superman engaged in a fight accompanied by lots of stimulation and as a result the cluster membership changes back, as shown in Fig. 3.11 D).

3.2.1 Conclusion: PDVC

This section developed the concept of positive developmental video classification. The work developed several features and a classification system that can be used to classify the content's impact on the cognitive, social and academic development of children. This work tests the model with a novel model validation procedure to ensure that classification is not based on similarities among series. Novel cognitive and structural features have been developed. The work finds that these features outperform the current features being used in video genre classification in the same modality. Additionally, music information retrieval features have been incorporated, such as spectral features and Pitch Chroma to capture audio characteristics associated with content that has a negative and positive impact. We also include affective features due to their link to cognition [2]. We investigate



Cluster 1

Cluster 2

Figure 3.9: Baby Einstein Top: Three images extracted from cluster one contain people reading text. Baby Enstein Bottom: Three images extracted for cluster, two showing individuals playing music and one showing a puppet accompanied by strange noises.

arousal features independently, explore the accuracy of the decision units over different modalities and explore how sampling rate impacts accuracy. Our results show that the problem of PDVC is different from the video genre classification task. We conclude that when the system is combined with classical features, the novel system has almost 30% better accuracy than the state-of-the-art system designed for the video genre classification task and 65% better performance than the ATC developed in [2].

3.3 Automatic Age-Recommendation System for Children's Video Content

This section develops a new method to determine age-appropriate content for children between the ages of three to six years old using global features. To our knowledge, there has been no previous research done on determining age specific content for children. TRECVID does not include age category determination for videos.

The novel features developed here use fundamental frequency, subharmonic-to-harmonic ratio (SHR) [111], and optimized zero crossing rate (OZCR). These are then used to count the number

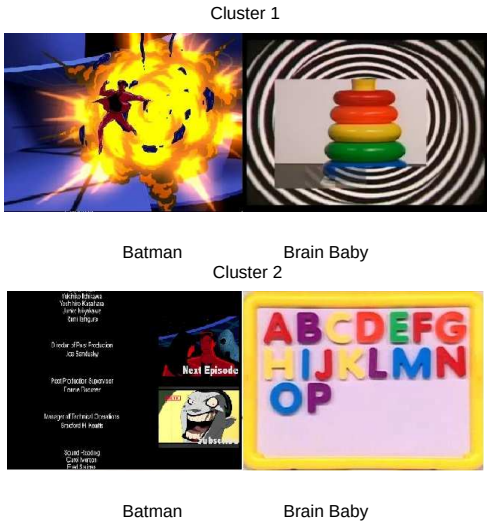


Figure 3.10: Top: Two images extracted from Batman and Brainy Baby that have some correspondence to the negative impact category. Bottom: Two images extracted from Batman and Brainy Baby that have some correspondence to the positive impact category.

of utterances of voiced speech. This information is then used as a means to determine language complexity. The novel algorithm is accurate and does not require training data or complex probabilistic inference and avoids the complexity of speech recognition. These cognitive capacity audio features help quantify the cognitive capacity of the intended target audience. The video test set is taken from an online commercial database for children. Finally, SVM is used as a classifier. Our experiments show that these new features can vastly improve upon classic video features. In addition, the novel audio features perform better when compared to classical features. Furthermore, the new feature extraction methods are computationally less expensive because we count the number of syllables and words using computationally inexpensive signal processing techniques without going through complex speech recognition.

3.3.1 Criteria to Classify a Video Into Different Age Categories

The age criterion has been determined a multitude of ways. For professionally produced videos age recommendation information has been included. For non-professionally produced videos, experts assign an age category using many factors including what kind of activity is being performed in

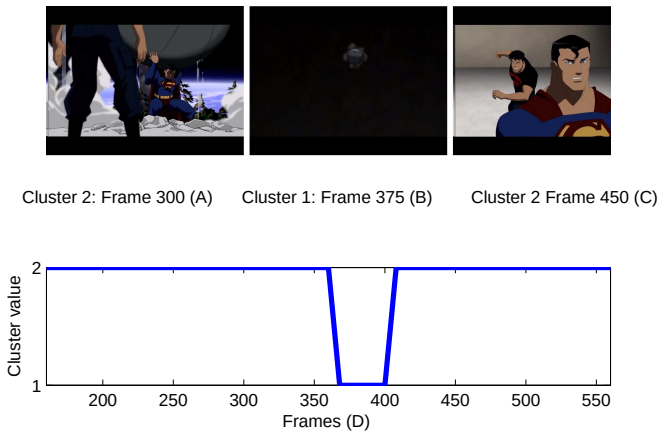


Figure 3.11: Top: Three images extracted from two different clusters from Batman the Animated Series, comprising of the 13-th Batman episode in the set consisting of Superman in a fight. Top: A) Superman lifts rock B) Superman throws rock in air C). Bottom: D) Contains cluster membership of each frame.

the video and who else has viewed the video. As the video database is geared for children, all the content is appropriate for children of all ages.

3.3.2 Novel Features: Automatic Age Recommendation

Novel Audio Features

Cognitive Capacity Audio Features

As children get older, the ability to process language undergoes a rapid change, and this is especially evident from the ages of three to six. In this section, we describe an algorithm to extract cognition-related audio features based on the number of words and syllables. Let $y(n)$ be the observed signal, let the speech portion be given by $s(n)$, where $s(n) = 0$ if the sample is not speech. Let ξ be the zero mean stationary noise term. Let the signal model be given by:

$$y(n) = s(n) + \xi(n). \tag{3.10}$$

The signal is normalized and the energy of the audio samples of the first frame of the video sequence are used to determine a noise floor. The audio segment is divided into intervals of thirty

seconds. The audio segment is then passed into the pitch determination module to determine the fundamental frequency. The signal is then partitioned into sub-signals with period N corresponding to a period of 40 milliseconds. Next we show that a threshold can be used to improve detection.

Let the samples of the ith sub-signals be denoted by (3.11):

$$\mathbf{Y}_i = \{y(iN+1), y((iN+2), ..., y(iN+N)\}. \tag{3.11}$$

Similarly, we can define $\mathbf{X}_i$ and $\boldsymbol{\xi}$ as sub-signals with speech and noise having the same relationship as in (3.11).

The model makes three assumptions: $1)$ the zero mean noise should be independent of the speech segment. Therefore, $cov(\mathbf{X}_j, \boldsymbol{\xi}) = 0$ where j is an arbitrary integer. $2)$ There is some correlation between speech segments $cov(\mathbf{X}_{i\in Z}, \mathbf{X}_{j\in Z}) = C_i$, where C_i is usually a positive constant and Z is the index set of speech. $3)$ Because our model assumes that $x(n) = 0$ then $cov(\mathbf{X}_{j\in\overline{Z}}, \mathbf{X}_i) = 0$. Using assumption 1, we can show that:

$$cov(\mathbf{Y}_i, \mathbf{Y}_{i+1}) = cov(\mathbf{X}_i, \mathbf{X}_{i+1}) + \sigma^2, \tag{3.12}$$

where σ^2 is the noise variance and the covariance is calculated with the samples in $\mathbf{Y}_i$. A threshold value can be determined using assumption 2 and 3:

$$\sigma^2 < cov(\mathbf{Y}_{i\in Z}, \mathbf{Y}_{i+1\in Z}) \tag{3.13}$$

The signal energy can also be used as a noise floor. Additionally, based on models of speech, if OZCR of either segment is higher than $3000Hz$ and SHR is between 0.2 and 0.4 [111], the frame is classified as voiced, i.e., a syllable. The block diagram is shown in Fig. 3.12 where i and $i+1$ represent two adjacent sub-signals.

If two syllables are more than 0.4 seconds apart, it is considered the start of a new word. Several additional filtering steps are also incorporated. The average number of syllables (ANS) and the average number of words (ANW) are then used as features. These measures are then used to calculate the novel measure of language complexity (LC). It is assumed that complex language has more syllables per word. Therefore the LC is defined as:

$$LC = \frac{Number\,of\,Words}{Number\,of\,Syllables}. \tag{3.14}$$

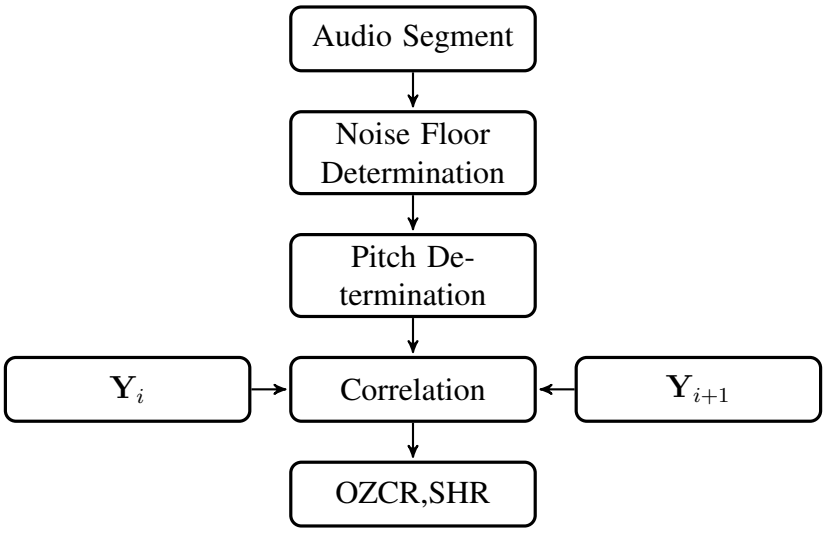


Figure 3.12: Block diagram of syllable decision algorithm.

As the value of LC gets smaller, the words contain more syllables and the language becomes more complex. As LC approaches one and language is considered simple, one-syllable words are used.

Average Number of Audio Spikes

It is observed that the audio of content for younger children does not have any sudden increases of amplitude. This is most likely because sudden spikes of audio would startle younger children. As a result, an algorithm to count the average number of audio spikes (ANAS) is developed. The algorithm is based on envelope detection, and initially, the short time energy (STE) is calculated. The bandwidth of the window is determined heuristically, then the short time energy is divided into K frames of size of 200 milliseconds. The total energy within a frame is then calculated, denoted by:

$$\Delta STE(k) = \sum_{m=k(M)}^{(k+1)M} \sum_{l=0}^{\infty} (y(l)w(m-l))^2, \tag{3.15}$$

where m denotes the sample number of the short time energy, k denotes the frame index, M is the frame size and w is the filter. By the central limit theorem, $\Delta STE(k)$ is normally distributed:

$\Delta STE(k) \sim \mathcal{N}(\mu_s, \varepsilon_s^2)$, where μ_s is the mean and ε is the standard deviation. If ΔSTE is below the 2.3 percentile, the short time energy for that frame index is set to zero.

$$\text{if } \Delta STE(k) < \mu_s + 2\varepsilon_s \text{ then } \Delta STE(k) = 0 \tag{3.16}$$

Finally, a peak-detection algorithm is used on the short time frame energy to count the peaks, and this number corresponds to the number of audio spikes.

Novel Motion Features

Average Color Histogram

Average color histogram differences (ACHD) is commonly used in shot boundary detection but is not usually used as a feature. In the present system, ACHD is used to quantify the amount of motion because it is not as susceptible to change in illumination as camera motion.

Video-Content Analysis Features

As well as the novel features, the new system uses some of the standard video content analysis features from the references referred to above. These features include a global RGB histogram (GRGBH) calculated with thirty-two bins. The first four color moments are used. The average is taken to determine the global color moments (GCM). Structural features are obtained using automatic shot boundary detection. These features include the average number of shots (ANS) and average shot length (ASL). Finally average pixel-wise differencing (APWD) is also used.

3.3.3 Classification

A total of 135 test videos are used from a commercial video streaming service geared for children. During testing, the audio-visual features are extracted from the input videos. Classification is performed using SVM. The features are extracted and the mean of the data is subtracted from all the data points and scaled with the standard deviation. The kernel functions used include: linear, quadratic, polynomial of order three and Gaussian radial basis function (RBF) kernels with a default scaling factor of one. In order to account for the multiple classes, the pairwise methods of selection are used.

Table 3.9: Actual vs Counted for number of words algorithm.

Video	Actual	Counted	Actual vs Counted
1	152	158	96.2%
2	145	143	101.1%
3	135	187	72.20%
4	155	187	82.90%
5	93	115	80.70%
6	110	116	94.80%

Table 3.10: Actual vs counted for number of syllables algorithm .

Video	Actual	Counted	Percentage
1	172	191	90.0%
2	203	203	100%
3	190	225	84.4%
4	211	198	107.0%
5	154	117	132.0%
6	130	134	97.0%

3.3.4 Experimental Results: Automatic Age Recommendation

Cognitive Capacity Audio Features

This section verifies the results of the novel features. The results of the approach for determining the number of words and syllables are displayed in Tables 3.9 and 3.10, respectively. The algorithm is tested from several TV shows in the test set. The actual results are set out in the column labelled *Actual* and the results from the algorithm are set out in the column labelled *Counted*. The average classification accuracy for number of words is 77.0% and for number of syllables is 97.0%.

Audio-Spike Detection Results

The results of the algorithm for detecting audio spikes are discussed and shown here. Fig. 3.13 (A) shows the time series obtained from an eight second recording of a conversation interrupted by two loud noises at the two and five second mark. Fig. 3.13 (B) shows the short-time frame energy. It is evident that the short-time energy is slowly varying compared to the time variations of the raw signal. This makes the peaks corresponding to the two loud noises much more apparent and easy

to find using a peak detection algorithm.

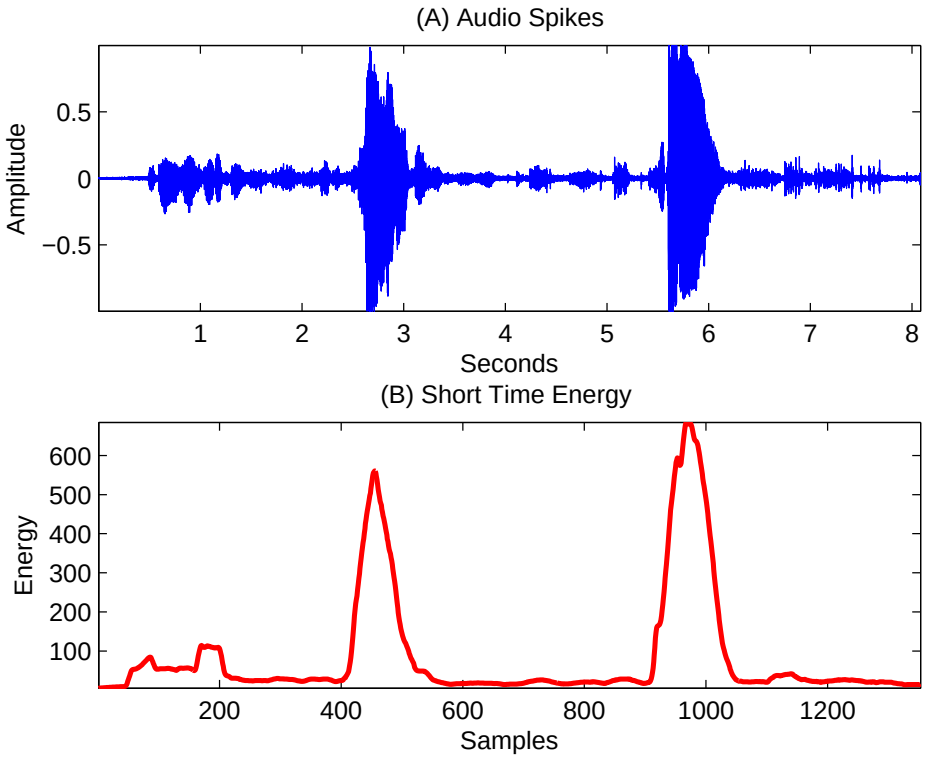


Figure 3.13: (A) A time series of a conversation interrupted by two loud noises (B)the short time frame energy of the time series above.

Table 3.11 shows the results of the audio spike detection algorithm. The second row shows the number of audio spikes and the third row shows the counted number of audio spikes. The algorithm's overall accuracy is just under 85%.

Age Classification Results

The video data is classified by several testers with knowledge in child psychology for appropriate age ranges between three and six years old. Experiments are performed on individual features, individual classes of features, overall classification accuracy, and different kernels. For each experiment k-fold cross-validation is performed. Cross-validation consists of randomly partitioning the set into two equal-size sub-samples. One of the sub-samples is retained as the validation data for testing the model, and the remaining sub-samples are used as training data. The cross-validation

Table 3.11: Actual vs counted for number of Audio Spike Detection Algorithm.

Video	Actual	Counted	Percentage
1	1	1	100%
2	4	3	75.0%
3	4	5	80.0%
4	4	5	80.0%
5	8	7	87.5%

Table 3.12: Comparison results accuracy of different kernels.

Kernel	Linear	Quadratic	Polynomial	RBF
Accuracy Of Old Features	79.0%	85.0%	85.0%	82.0%
Accuracy Of New Features	79.0%	87.0%	85.0%	79.0%
Combination of All Features	**93.0%**	92.0%	92.0%	85.0%

process is then repeated with each of the sub-samples used exactly once as the validation data. Once the classification is performed, the results are then averaged to produce a single estimation. Average correct classification rates over these 60 experiments are presented.

First, we compare the results of the novel features compared to the classical features and total features as shown in Table 3.12, where the rows correspond to different kernels. It is evident with the exception of linear kernels and RBF that the novel features outperform the classic features. The system works better using all the features with a maximum increased accuracy of 13.0% for the same classifier and an average improvement over all classifiers of 75.6%.

The linear kernel performs better as the dimension of the data increases, and this is because as the dimension of the data increases, the data becomes linearly separable. Furthermore as the dimension of the data increases, other kernels become susceptible to over-fitting.

The confusion matrix shown in Table. 3.13 is used to further explore our results. Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class. We only display the confusion matrix for the experiment using all the features and only with the linear kernel because it performs best. The most difficult age group to classify correctly is found to be age three, and this age group is most often confused with age four. Further investigation reveals that many of the misclassified samples are difficult even for our experts to

Table 3.13: Confusion matrix for linear kernel.

Age	3	4	5	6
3	6.6	3.3	0	2.1
4	3.3	38.7	0	0
5	0	0	52.2	1.8
6	0	0	0	27

Table 3.14: Accuracy of different class of features and different kernels.

Kernel	Linear	Quadratic	Polynomial	RBF
Motion	63.0%	73.0%	72.0%	**76.2%**
Color	75.0%	**84.0%**	81.0%	79.0%
Structural	40.0%	43.0%	44.0%	42.0%
Audio	74.0%	85.0%	**85.0%**	**80.0%**

accurately assess.

Classification accuracies for audio, motion, color, and structural features are shown in Table 3.14, where each row corresponds to a different group of features. We see that the novel audio features have superior performance with an accuracy of 85.0%. The color features also perform well with a maximum accuracy of 84.0%. This is most likely due to the color feature vector being considerably larger than any other feature. Motion features also perform well with a maximum accuracy of 76.0%.

Subjective Evaluation

Using subjective evaluation we see a marked difference in the videos. For example consider Fig. 3.14, a screen shot of a video for children three years of age. The video consists of an individual reading a book about the adventures of a bird. Compare this to the video for six year olds, with a screen shot shown in Fig. 3.14, wherein the video gives children complex instructions on how to make ”Magic Mud”.

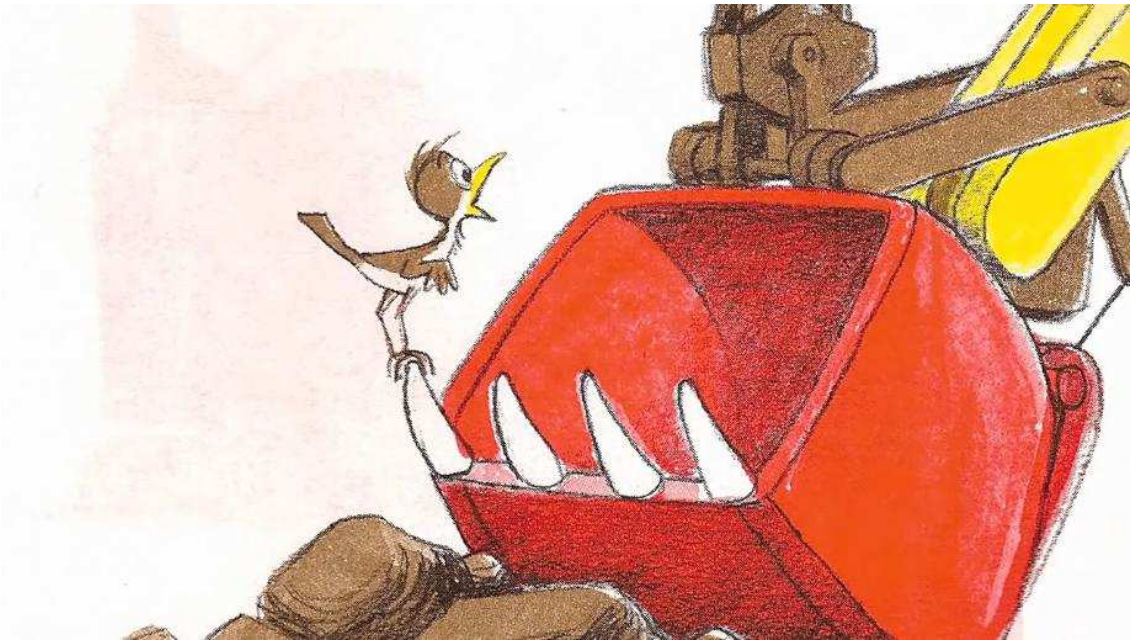


Figure 3.14: Example of content for children three years of age: simply an individual reading a book on YouTube.

Figure 3.15: Example of content for children six years of age: Instructions how to make ”Magic Mud” om YouTube.

Conclusions: Automatic Age Recommendation

In this section, we have presented an automatic method of determining the appropriate age category for a video database built for children. The novel system exploits a combination of motion, color, and structural features as well as several novel audio features that gauge the cognitive abilities of the intended audience. This is the first time, to our knowledge, that this type of video classification has been performed. The audio features include the average number of syllables, words and sound jumps. New efficient signal processing algorithms are developed to extract these new audio features related to the language structure and complexity, and these novel audio features perform better than the standard video features. This approach is extensively tested on 93 hours of video. Each video is labelled with a recommended age of three to six years. The method is tested using K-fold cross-validation and SVM is used as the classifier. The experiments are performed on different sets of features. It is observed that the system has an accuracy of 92.2% using all features, and the novel features perform better when tested against classic features and improved performance up to 13.0%.

Chapter 4

Dynamic Time-Alignment K-Means Kernel Clustering For Time Sequence Clustering

In the previous chapter, we demonstrate that features that correspond to heightened levels of arousal are extremely accurate in predicting whether content has a negative or positive impact on cognition. Furthermore it has been established that certain ranges of valence can be used to predict the impact on cognition. These results have been predicted in the physiological literature [10, 11, 12, 13, 14, 15, 16]. As a result, this chapter develops a method to rank time series by using a combination of kernel clustering [112, 113] and dynamic time alignment kernel clustering [114]. The contributions to our knowledge include a novel method to rank time series using clustering, experiments in ranking and clustering time series with respect to valence and arousal, and a novel transformation that incorporates prior knowledge of the valence arousal plane into the cluster assignments.

4.1 Introduction

The main problem is how to rank valence and arousal with variable granularity. This is accomplished using clustering, but this still leaves the question of how to effectively rank sequences. Therefore, this section presents a novel method of clustering sequences by embedding a non-linear time alignment kernel function into kernel k-means. The time-alignment operation embeds the sequential pattern in the kernel function, allowing kernel k-means to be used to classify entire sequences. The method is evaluated with over 9800 videos and with features from the LIRIS annotated creative commons emotional database. A Matlab implementation of the novel algorithm is

available at [115]. Using a greedy kernel cluster algorithm with a novel algorithm to generate toy data for testing, our results show that the method works well in classifying sequences based on their affective content and performs better than other unsupervised methods for clustering time series. The linking transformation is also developed and acts as a method to incorporate prior knowledge into the cluster assignment. This dissertation also evaluates the ability of several methods to map low-level features onto the valence-arousal plane from the LIRIS database. Clustering on the valence and arousal axis is performed independently, the results are verified with the rankings from the LIRIS database. When both valence and arousal are used, the results are verified heuristically. The regression results also show that simple ridge regression has comparable performance to state-of-the-art regression methods.

A time series of features is extracted from a video sequence and mapped to the valence arousal plane. The main objective of the method is to perform a novel clustering method on a set of movies, then cluster the entire movie sequence. Each cluster represents movies with similar emotional content.

As stated in Chapter 1, affective tagging for video describes the emotional content of a video and has many applications, [66, 64, 116]. The most direct representation of an emotion is to use discrete labels or emotional prototypes. Examples include fear, anxiety, and joy. This method has many problems: labels are not universal, labels can be misinterpreted and emotions are a continuous phenomena rather than discrete [64]. Finally, fixed classes can only be changed by combining or splitting certain classes to reduce or increase the emotional granularity [69, 70].

Another method is to use the 2-D emotion space to describe the emotional content of a video sequence [2]. This space contains the valence dimension and the arousal dimension. Valence describes the type of emotion: negative to positive. Arousal describes the intensity: inactive to active. Any point on this space can be used to describe different emotions. It has been shown that low-level video features can be mapped onto this space using regression [66]. An example of the 2-D emotion space with some discrete labels is shown in Fig. 4.1 and with different time curves generated in Fig. 4.2 by a video representing the emotions elicited. The parabolic shape is due to the relative low number or even totally absence of stimuli that would cause a sudden change in an

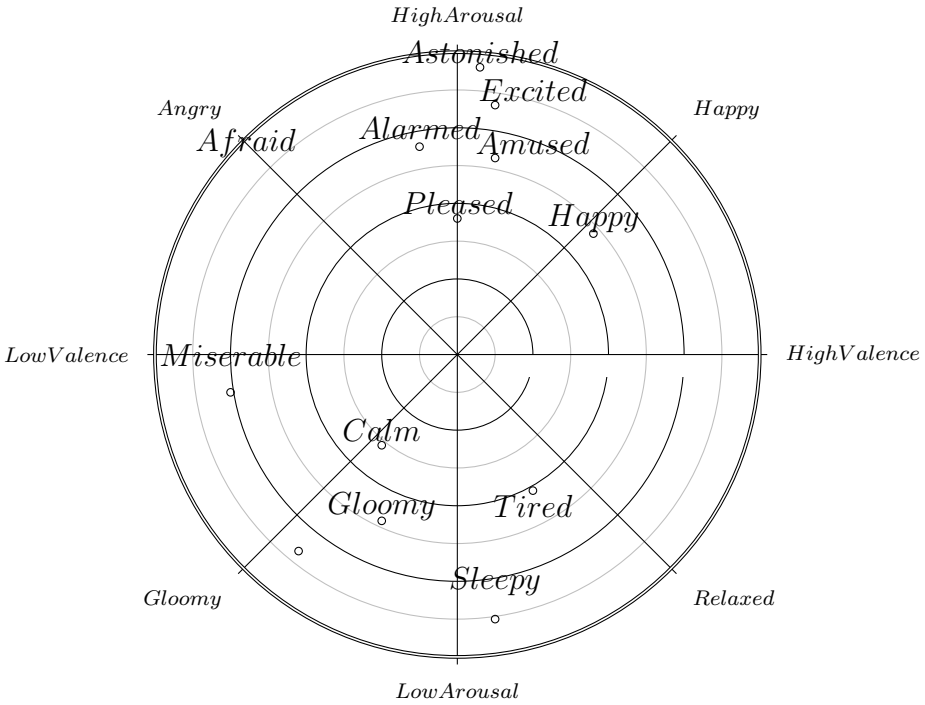


Figure 4.1: 2-D emotion space; the diagonal axis represents valence, and the horizontal axis represents arousal. In addition, there are several discrete labels corresponding to different emotional prototypes.

emotional state, for instance, high arousal and neutral valence, or high valence accompanied by low arousal [2].

Different individuals have different curves but most of the variation is in the lower region [66]. This is demonstrated in Fig. 4.3, as in both videos the lower region is linear for one individual and more quadratic for the other individual. In the upper region both curves appear similar, but there is usually some variation between curves from different individuals.

The curve also faces the granularity problem. The continuum has to be quantized again to produce possible system responses [69]. Consider the ranking problem of classifying content into low, medium, or high arousal and valence categories [66]. If the user decides to change the granularity of the system, new labels must be obtained and the system must be retrained. Therefore several methods have used clustering as a means of segmenting the data [69, 117]. The main problem with these methods is they do not handle time series data. The issue with time series is

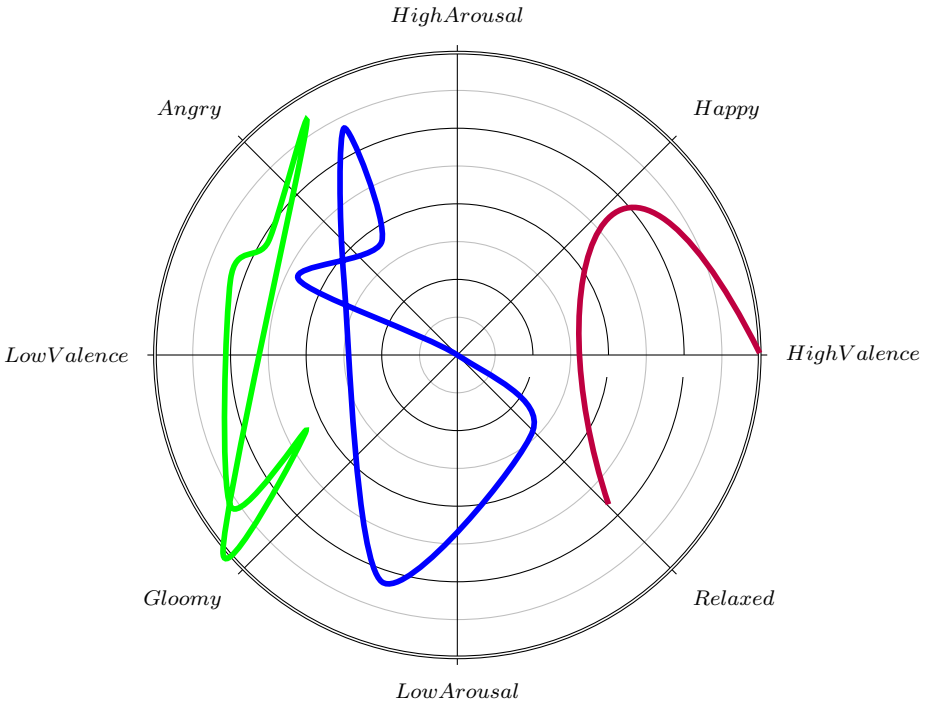


Figure 4.2: 2-D emotion space: The diagonal axis representing valence, the horizontal axis representing arousal, and three time curves generated by three video sequences.

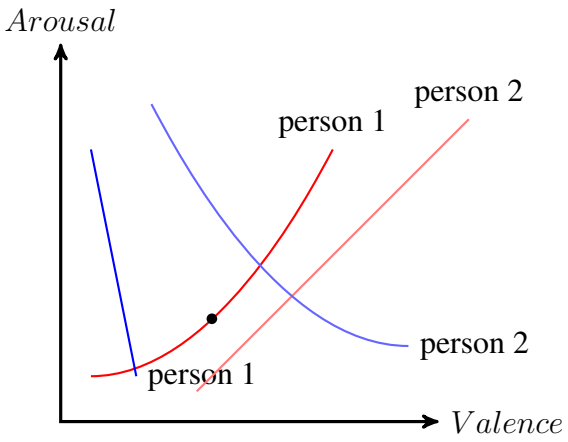


Figure 4.3: Two-dimensional valence arousal time series generated by two people for two different video clips, clip one is in blue, clip two is in red. There is more variation in the lower region, but the curves become similar in the higher region.

that it may not be localized in one area. An example of three time series is shown in Fig. 4.2, and it is not difficult to see that one can rank the time series into a low, medium, and high valence level, but low and high ranking are also feasible. The feasible rankings are shown in Fig. 4.4, wherein the first level shows the 2-D emotion space with three time series, the top shows the time series ranked into low and high arousal valence levels, and, finally, above we see the ranking of the different time series into low, medium, and high valence levels.

The problem with using a direct feature vector representation of time series is the curse of dimensionality, i.e., large time series require lots of training data. In addition, in many applications such as this, time series are of different lengths. Therefore, this section develops dynamic time-alignment k-means kernel clustering (DTAKKC). The method uses dynamic time-alignment kernels [114] that approximate the kernel operations of a series of different lengths with a dynamic time warping kernel, then spectral kernel k-means [112, 113] is used to cluster the data. Kernels are ideal for clustering because they are more robust to high dimensional data. The method outperforms dynamic time warping, k-means (DTK) [118], as well as wavelet histogram methods (WHM) [119].

Another issue that has not been tackled with respect to time series is incorporating prior knowledge into the cluster. Consider the problem of violence annotations of affective classes [120], or scary scene detection [121]; we would like to keep content with those elements in the same clusters. This is demonstrated in Fig. 4.5; the shaded region corresponds to emotions that should be segmented in the same cluster. Series that contain samples in that region are in red and should be assigned in the same cluster even though they may appear to have more in common with series in the second cluster coloured in green. In this example there are only two clusters but the problem could be extended to more clusters.

The main advantage of this kernel method includes:

- better generalization performance on experimental data
- better heuristic performance
- one can design new kernels to improve performance

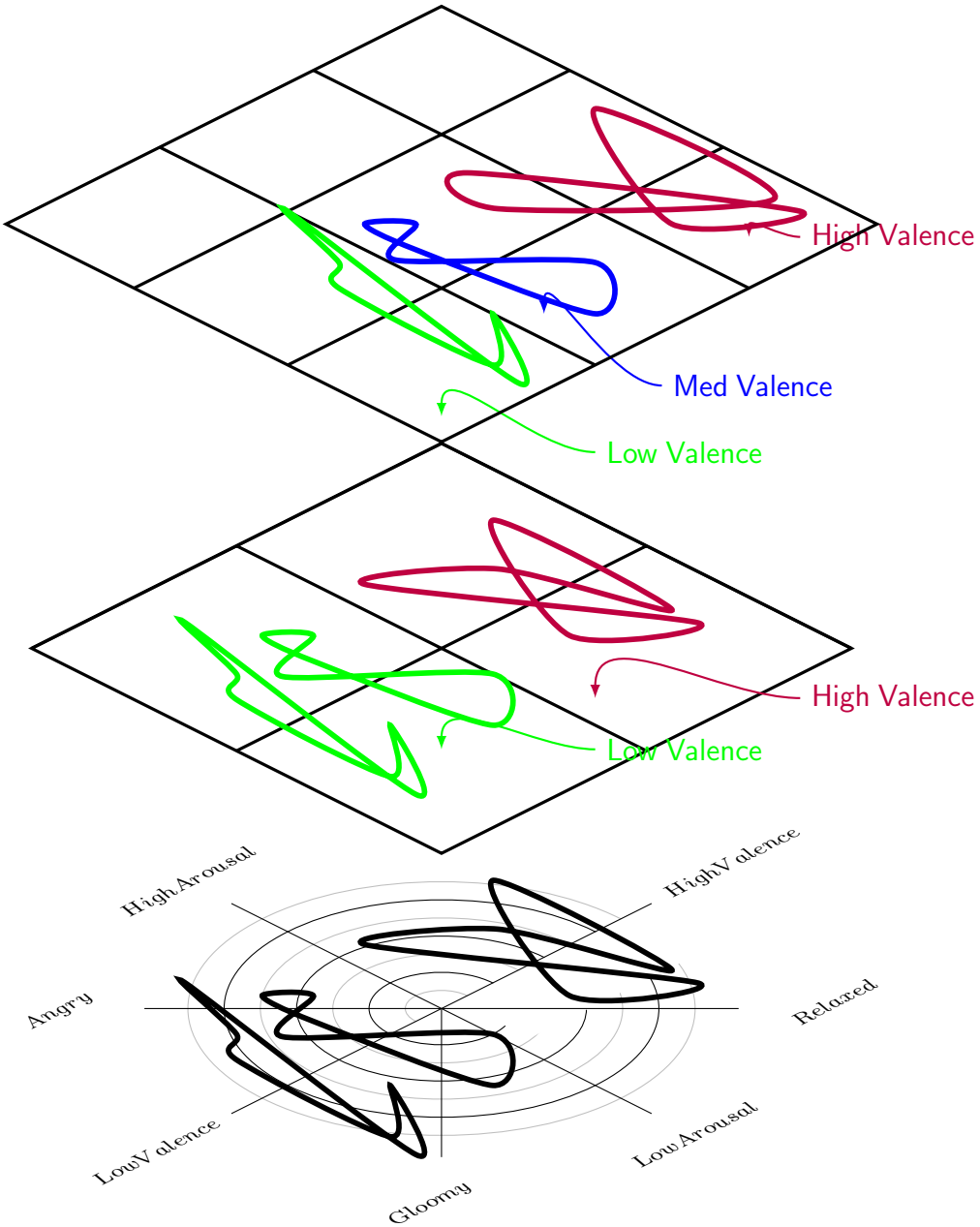


Figure 4.4: Example of ranking time series using different quantization of the valence axis. *First level*: three time series on the 2-D emotion space. *Second level*: ranking of time series into low and high arousal valence. *Third level*: ranking of time series into low and high arousal valence third level.

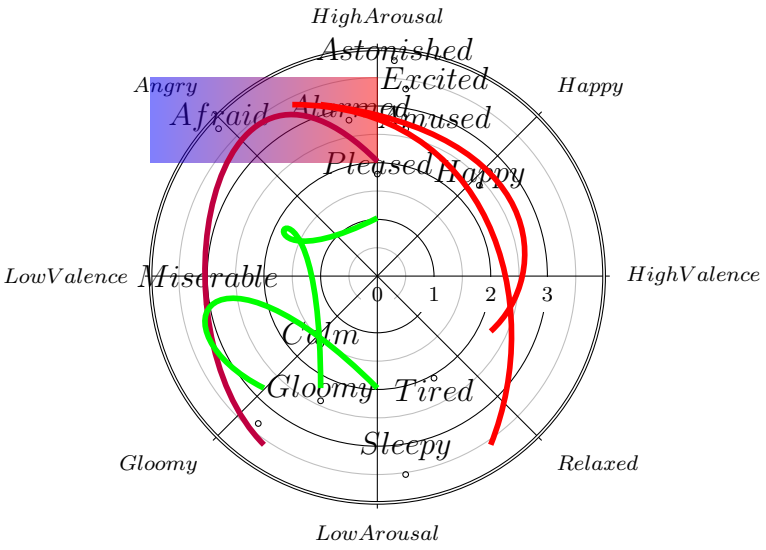


Figure 4.5: Representation of VA plane with region that contains emotions that should be segmented in the same clusters. Series that contain samples in that region are in red and assigned in the same cluster. The remaining clusters are in green.

- finding non-linear patterns at a reasonable computational cost (RBF kernel)
- limited to no free parameters
- transformations that use prior knowledge

The method is tested with features and ratings from LIRIS-annotated creative commons emotional database (LIRIS database) for affective video-content analysis [122, 123, 124]. In addition to developing the novel algorithm, we also test several regression methods on mapping the features that have not been tested on the LIRIS database. These methods include the most popular methods used in affective analysis [64, 66, 65, 34], including relevance vector machines (RVM) [125], ridge regression (RR) [90], and neural networks. Finally, we develop the linking transform as a means of incorporating prior knowledge into the cluster assignment.

4.2 Problem Formulation

4.2.1 Regression Problem

Due to the different types of regression methods, we formulate the problem using empirical risk minimization [83]. Let $U = \{v, a\}$ where a indicates arousal and v indicates valence, let $u \in U$ be the variable indicating the type of features and ratings used. Let $\mathbf{x}_{n_c,u}$ be the feature vector from a training set consisting of N_c videos clips, with annotation $\Gamma_{n_c,u} \in \Re$ be the type of rating. The goal for each rating is to determine a hypothesis that minimizes the risk:

$$\hat{h}_u^* = \underset{h_u \in \mathcal{H}}{\operatorname{argmin}} \{ \frac{1}{N_c} \sum_{n_c=0}^{N_c} \mathcal{L}(h_u(\mathbf{x}_{n_c,u}), \Gamma_{n_c,u}) \}, \tag{4.1}$$

where $\hat{h}_u$ is the hypothesis, $\mathcal{L}(\cdot)$ is some loss function and $\mathcal{H}$ is the set of all hypotheses. Once the optimum $h_u(\cdot)$ has been discovered we can map different features into the valence and arousal space (AS):

$$\hat{\mathbf{y}}_{n_c} = [h_a(\mathbf{x}_{n_c,a}), h_v(\mathbf{x}_{n_c,v})]^\top. \tag{4.2}$$

4.2.2 Algorithm

Let $\hat{Y}_i = [\hat{\mathbf{y}}_{1i}, ..., \hat{\mathbf{y}}_{N_i i}]$ be the predicted points of the AS for video sequence i of length N_i and $\hat{Y}_j = [\hat{\mathbf{y}}_{1j}, ..., \hat{\mathbf{y}}_{N_j}]$ be the predicted points of the AS for video j of length N_j. The Dynamic Time-Alignment Kernel is given by:

$$\mathcal{K}(\hat{\mathbf{Y}}_i, \hat{\mathbf{Y}}_j) = max_{\psi,\theta} \{ \frac{1}{K_{\psi,\theta}} \sum_{k=0}^{K} q(k) \kappa(\hat{\mathbf{y}}_{\psi(k)j}, \hat{\mathbf{y}}_{\theta(k)i}) \}, \tag{4.3}$$

where $\psi(k)$ and $\theta(k)$ are the time warping functions, $K_{\psi,\theta}$ is the normalizing factor, K is the length of the sequence, $q(k)$ is the path weighting coefficient and $\kappa(\cdot)$ is the kernel. Once the kernels have been calculated, kernel clustering can be used to determine the class membership using the following equation:

$$\max_{\{C_m\}} \sum_{m=1}^{M} w_n \sum_{\hat{\mathbf{Y}}_i, \hat{\mathbf{Y}}_j \in C_m} \mathcal{K}(\hat{\mathbf{Y}}_i, \hat{\mathbf{Y}}_j). \tag{4.4}$$

The term w_m is the clustering normalizing constant, and C_m is the cluster membership. We use the notation $C_{m,n}$ to represent the assigned label of each N video sequence samples. Both the relaxed and greedy methods of kernel clustering are used [126]. It is also useful to define the kernel matrix $\mathbf{K}$ of the training set, with elements $(\mathbf{K})_{i,j} = \mathcal{K}(\hat{\mathbf{Y}}_i, \hat{\mathbf{Y}}_j)$.

4.2.3 Novel Linking Transformation

To incorporate prior knowledge or modify the clusters assignment without recalculating the dynamic time-alignment kernel, we introduce a transformation that will be referred to as a linking transformation. These transformations are dependent on the linking matrix comprised of linking functions. Consider $l-th$ linking matrix, for the $i-th$ series the linking function is given by $f_l(\hat{\mathbf{Y}}_i) \in \Re$. The linking matrix is a diagonal matrix given by:

$$\mathbf{F}_l = \text{diag}([f_l(\hat{\mathbf{Y}}_1), \ldots, f_l(\hat{\mathbf{Y}}_N)]) \tag{4.5}$$

The linking transformation can be applied to the kernel matrix as follows:

$$\hat{\mathbf{K}}_L = \prod_{l=1}^{L} \mathbf{F}_l \mathbf{K} \mathbf{F}_l \tag{4.6}$$

Each element of the transformed kernel matrix is given by:

$$(\hat{\mathbf{K}}_L)_{i,j} = \prod_{l=1}^{L} f_l(\hat{\mathbf{Y}}_i) \mathcal{K}(\hat{\mathbf{Y}}_i, \hat{\mathbf{Y}}_j) f_l(\hat{\mathbf{Y}}_j) \tag{4.7}$$

If the series $\hat{\mathbf{Y}}_i$ and $\hat{\mathbf{Y}}_j$ exhibit some similarity not captured by the kernel, the values of the linking functions $f_l(\hat{\mathbf{Y}}_i)$ and $f_l(\hat{\mathbf{Y}}_j)$ can be determined such that $|(\hat{\mathbf{K}}_l)_{i,j}| > |(\mathbf{K})_{i,j}|$ making these series more likely to appear in the same cluster. The admissibility conditions of a kernel that have been met by the Dynamic Time Alignment Kernel have been shown in [114]. In the Appendix we show that if $\mathbf{K}$ satisfies Mercer's theorem [96] then for all l if the $rank(\mathbf{F}_l) = N$ then $\hat{\mathbf{K}}_L$ also satisfies Mercer's theorem. In the next subsection, we will introduce the linking functions used, we will drop the l to simplify notation.

4.2.4 Linking Functions Used

As certain regions on the VA plane have importance, such as scary scenes, we introduce the region segmentation linking function. The linking function multiplies the kernel with a constant $\mathcal{C}_1$ if any of the samples of the sequence are in a specified region $\mathcal{R}_{eg}$. If the sequence has no values in the specified area, the sequence is multiplied by the reciprocal. The region segmentation linking function is given by:

$$f_1(\hat{\mathbf{Y}}_i) = \mathcal{C}_1 I((\hat{\mathbf{Y}}_i)_k \in \mathcal{R}_{eg}) + \frac{1}{C_1} I((\hat{\mathbf{Y}}_i)_k \notin \mathcal{R}_{eg}) \tag{4.8}$$

Where the $(\cdot)_k$ indicates any sample from that series. It is not difficult to show that if $\hat{\mathbf{Y}}_i \in \mathcal{R}_{eg}$ and $\hat{\mathbf{Y}}_i \in \mathcal{R}_{eg}$ then the kernel linking transformation $(\hat{\mathbf{K}})_{i,j} = {\mathcal{C}_1}^2 \mathcal{K}(\hat{\mathbf{Y}}_i, \hat{\mathbf{Y}}_j)$. If $\hat{\mathbf{Y}}_i \notin \mathcal{R}_{eg}$ and $\hat{\mathbf{Y}}_i \notin \mathcal{R}_{eg}$ then $(\hat{\mathbf{K}})_{i,j} = {\mathcal{C}_1}^{-2} \mathcal{K}(\hat{\mathbf{Y}}_i, \hat{\mathbf{Y}}_j)$. Finally if one of the samples is in the $\hat{\mathbf{Y}}_i \notin \mathcal{R}_{eg}$. For the remainder of the cases, it is not difficult to show that $(\hat{\mathbf{K}})_{i,j} = \mathcal{C}_1 \mathcal{K}(\hat{\mathbf{Y}}_i, \hat{\mathbf{Y}}_j)$. In this thesis the value was selected as follows:

$$\mathcal{C}_1 = max_{i,j}((\mathbf{K})_{i,j}) \tag{4.9}$$

4.2.5 Block Diagram

The final block diagram is illustrated in Fig. 4.6. The right side of the figure shows the training and pre-processing steps. Mapping to the valence arousal plane is performed using a supervised regression algorithm denoted with the green block. The linking function can be determined using a training algorithm but for this work the values are determined using the pre-specified criteria of any point in the series passing through regions of the valence arousal plane that contain violence or a scary scene. The linking functions are then converted to matrix form, denoted by the second green block on the right side. The left side of Fig. 4.6 shows the actual algorithm, the purple block on the top of the figure represents feature extraction. The feature vectors are mapped to the valence arousal plane denoted by the next block. The block below shows the dynamic time alignment kernel calculated between two time series. Finally if necessary the linking transform is applied before clustering.

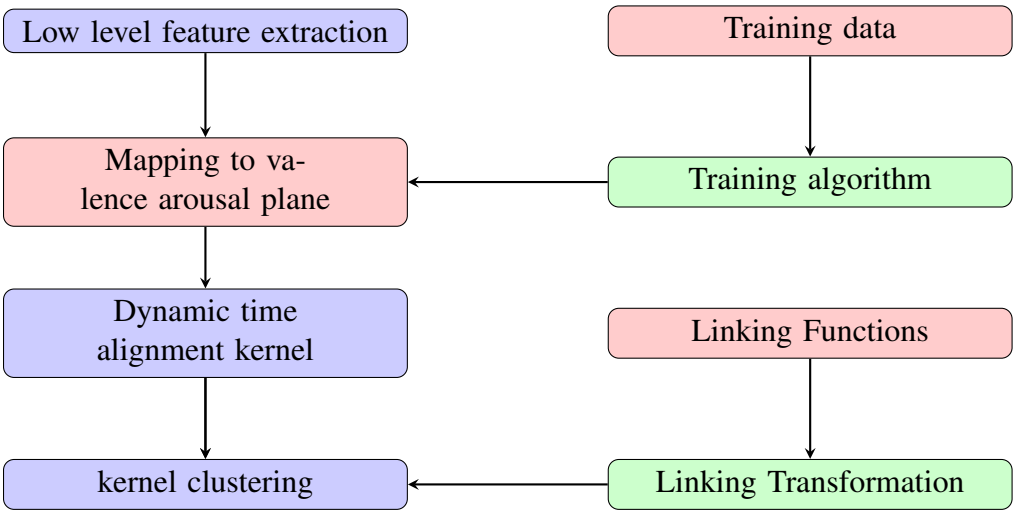


Figure 4.6: Block diagram of process: right side pertains to training and testing, left side represents process for sequence mapping and clustering

4.3 Procedure

4.3.1 Database

The method is tested with the LIRIS database for affective video-content analysis [122, 123, 124]. The database contains 160 films and short films. Different genres are used and segmented into 9800 video clips and one feature vector is extracted from each clip. The total time of all 160 films is 73 hours, 41 minutes and 74 seconds. The segmentation of each clip has been done using robust cut and fade in/out detection [127].

The best temporal resolution is still open, and as the work is constrained most experts agree that a few seconds is ideal [128, 129, 130, 131]. Additionally, [131] has shown that global ratings of perceived emotion for movies lasting a few minutes are not simple averages over time, but rather are more influenced by highly arousing events. As a result each clip is between 8 and 12 seconds.

The 23 arousal features $\mathbf{x}_a$ are available in the database and have been used to predict arousal. The valence features $\mathbf{x}_v$ are from both 2013 and 2015 sets. The valence and arousal values are for $y_{i,a}$ and $y_{i,v}$ respectively. After the values for valence and arousal, the samples $\hat{\mathbf{y}}_m$ have been concatenated into $\hat{\mathbf{Y}}_i$ for clustering. The list of features used is given in Table. 4.1.

Table 4.1: List of features used in LIRIS-ACCEDE Database, * indicates features found in the database with no reference.

Arousal Visual Features
Global activity (GA), Number of Scene Cuts Per Frame (NSPF), Median lightness (ML), Lighting, Colorfulness[132] Length of scene cuts (LSC),Harmonization energy (HE) [133]
Arousal Audio Features
Standard deviation of the i-th wavelet coefficients(DW_i))[34],Audio flatness envelop (AFE), Slope of power spectrum (SPS)
Valence Visual Features
Colorfulness [132],Hue count (HC) [134],Disparity of most salient points (DMSP),Depth of field(DF) ,Compositional balance (CB) [135]
Valence Audio Features
Zero-crossing rate (ZCR),Entropy complexity (EC) ,Asymmetry envelop (AE),Flatness Envelope (FE)
Other Features
Min Energy (ME), Frequency Centroid (FC), Maximum salient pixels Count (MSC) edge distribution area or Spatial Edge Distribution Area (SEDA),Alfa [136] Standard deviation of local maxima (SDLM)[122] ,Asymmetry,Spectral Slope(SS), White Frames (WF), Color Strength*(CS), Color Raw Energy *(CRE)

4.3.2 Model Validation Regression

The data set has been partitioned randomly using cross-validation and experiments performed 30 times and averaged. The values of the free parameters are determined using the validation set and the results on the test set (TS), with the training set denoted by TR. For the clustering, all the data has been used. The squared correlation coefficient R^2 has been calculated on the training data, and the predictive leave-out squared correlation coefficient Q^2 [123] is also used and both are given by:

$$R_u^2 = 1 - \frac{\sum_{n\in \mathrm{TR}}(\Gamma_{n,u} - h_u(\mathbf{x}_{i,u}))^2}{\sum_{n\in \mathrm{TR}}(\Gamma_{n,u} - \hat{\mu}_u)} \tag{4.10}$$

$$Q_u^2 = 1 - \frac{\sum_{n\in \mathrm{TS}}(\Gamma_{n,u} - h_u(\mathbf{x}_{n,u}))^2}{\sum_{n\in \mathrm{TS}}(\Gamma_{n,u} - \hat{\mu}_u)}, \tag{4.11}$$

Where TS is the test set and $\hat{\mu}_u$ is the empirical mean of the training data, excluding the testing and validation samples. We also determine the residual squared error of the test set. It should be stressed that the term $(\hat{\mu}_u)$ is from the training data and not the testing data, hence the denominator may be larger than the energy of the signal. The residual squared error is also calculated and for this section is given by:

$$RSE_u = \sum_{n\in \mathrm{TS}} (\Gamma_{n,u} - h_u(\mathbf{x}_n))^2 / \sum_{n\in \mathrm{TS}} \Gamma_{n,u}^2. \tag{4.12}$$

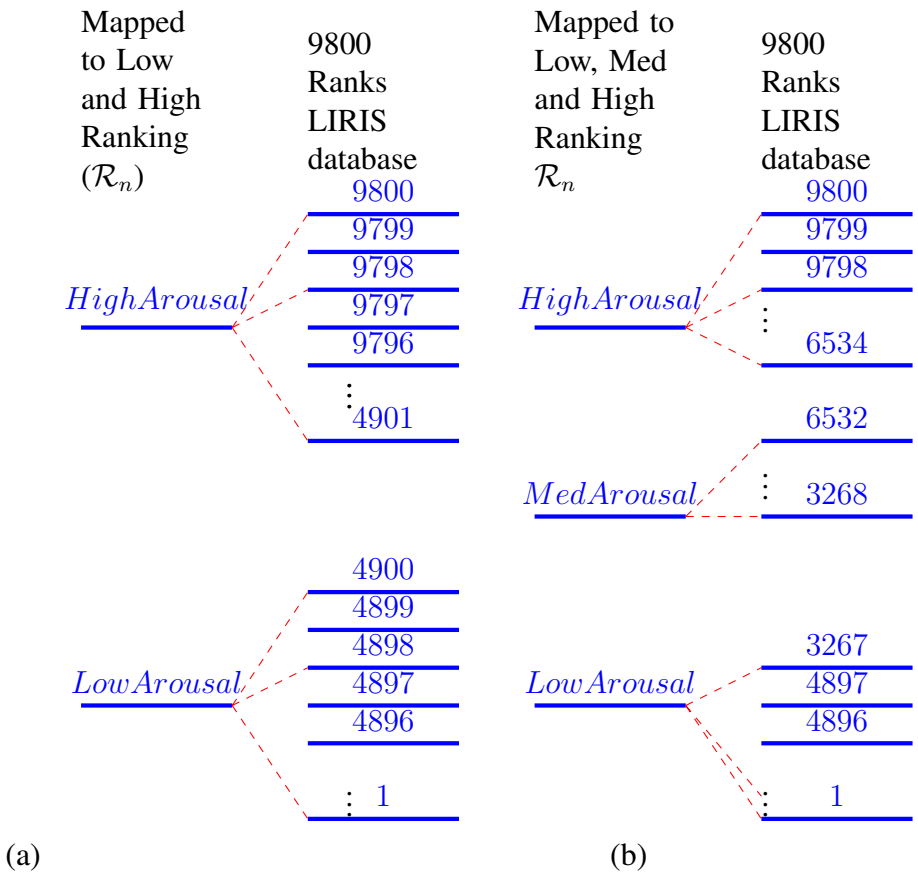


Figure 4.7: a) Illustration of 9800 rankings from LIRAS database quantized into two rankings, b)Illustration of 9800 rankings from LIRAS database quantized into three rankings

We also perform bias variance decomposition on the best performing models coupling the bias and noise into one term. Although features are relatively inexpensive and the data set relatively small, feature selection has been performed using Elastic Net and Lasso.

4.3.3 Validation Clustering

Unfortunately, global rankings for each sequence does not exist. As such, several methods for determining clustering performance have been used: the valence and arousal rankings from the LIRIS database are used by quantizing the rankings to a different granularity. The process is shown in Fig. 4.7 where the 9800 rankings are quantized into two levels and three levels respectively.

For valence and arousal, the quantized rankings from the LIRIS database are used as a ground

truth and compared to the unsupervised method. An illustration of the validation process is shown in Fig. 4.8, and three time series are automatically ranked according to low, medium, and high valence in green, blue, and purple respectively. The bottom axis represents the quantized rankings in the LIRAS database. The red portions of the time series are those that are misclassified by the unsupervised method. Let R_n be the correct ranking label, the average accuracy is given by:

$$\frac{1}{N}\sum_{n=1}^{N} I[C_{m,n} = \mathcal{R}_n]. \tag{4.13}$$

The accuracy per series is the number of correct classifications in each time series. The accuracy of series j is given by:

$$ACS_j = \frac{1}{N_j}\sum_{n\in S_j}^{N_j} I[C_{m,n} = \mathcal{R}_n], \tag{4.14}$$

where S_j is the index set of the $j-th$ series of length N_j. The average accuracy per sequence is also used and is given by:

$$\frac{1}{N_s}\sum_{j=1}^{N_s} ACS_j, \tag{4.15}$$

where N_s is the number of sequences.

In addition to the labeling, performance is also determined using visual assessment on how the cluster time series are constrained within a specific region of the valence arousal plane, and this represents the average range of emotions.

4.4 Results

4.4.1 Regression Results

The regression results for valence and arousal prediction are shown in Table. 4.10. The $\times$ is due to singular design matrix. It is evident that the values of R^2 and Q^2 provide little information. Most methods have an RSE of 4% for valence and 12% for arousal. It is evident that no method performs substantially better than ridge regression. These statistics correspond to very strong positive linear relationships; in the case of valence we see an almost positive linear relationship.

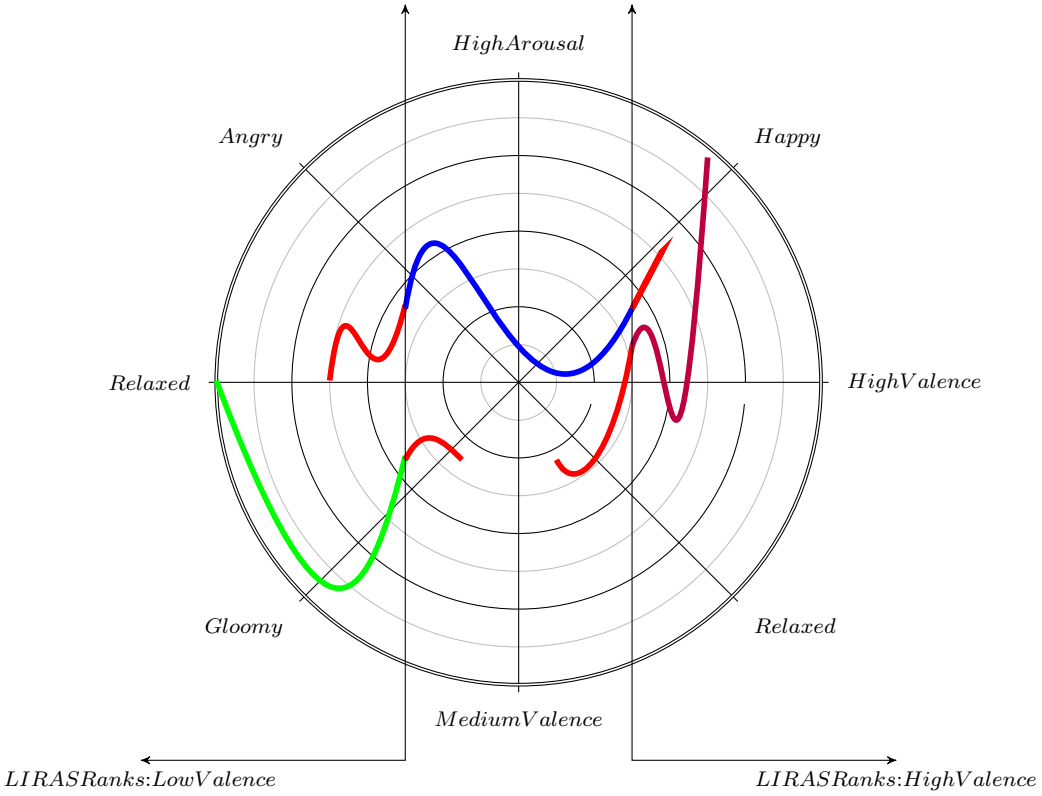


Figure 4.8: An illustration of the validation process, three time series are automatically ranked according to low, medium and high valence in green, blue and purple respectively. The bottom axis represents the quantized rankings in the LIRAS database. The red portion of the time series are those that are misclassified by the unsupervised method.

This is in contrast to other work that has found arousal to have a stronger correlation with low-level features [66]. The good performance of RR suggests Gaussian noise and a linear relationship. Finally, we include a two-layer feed forward neural network, with all free parameters having been selected using Matlab's built-in functions.

Table 4.2: Results of regression methods Var(RSE) indicates empirical variances of the RSE and $\times$ indicates regularization error

Method	R^2	Q^2	RSE	Var(RSE)
Liner Regression	($\times$,1)	(0.998,0.998)	($\times$,12.1759%)	($\times$,1.22e-5)
Ridge Regression	(1,1)	(0.998,0.998)	(4.1623%,12.1751%)	(5.41e-6,1.51e-5)
RVM	(1,1)	(0.998,0.998)	(4.1621%,12.1748%)	(4.24e-6,1.48e-5)
Neural Network	(1,1)	(0.998,0.998)	(4.1632%,12.1757%)	(3.67e-6,2.39e-5)

The Lasso results are shown in Table. 4.10, Elastic Net performs better than the Lasso as expected, as it includes correlations between variables, but there is no substantial improvement. Fig. 4.9 shows the average RSE for different values of the regularization term using cross validation, a value of $10^{-3.6}$ indicated by the green curve has the lowest RSE and 10^{-2} in blue has the most zero values of $\mathbf{w}$. Fig. 4.10 shows the value of the different parameters w_j for different values of the regularization parameter, with the green and blue vertical lines having the same meaning as Fig. 4.9. RR is also used for predicting ranking for valence and arousal in order for the results to be compared to [34]. The error for both is approximately 23.00%, with the similarity between the two measures found in [34]. The RSE found in [34] is about 32.00%, larger than the values found here. This is probably due to a much more constrained validation process that leads to less training data.

Table 4.3: Results of regression methods Var(RSE) indicates empirical variances of the RSE and $\times$ indicates regularization error

Method	R^2	Q^2	RSE	Var(RSE)
Ridge Regression Lasso	(1,1)	(0.998,0.998)	(4.21%, 12.14%)	(0.4957e-5,1.51e-5)
Ridge Regression Elastic Net	(1,1)	(0.998,0.998)	(4.161%,12.07%)	(5.41e-6,1.418e-5)

The optimum value for the parameters w_j for arousal and valence are given in Fig. 4.11 to Fig. 4.14. Examining Fig. 4.11 we see that using lasso, some of the valence features have prediction ability on arousal. Examining Fig. 4.12 we see that by using Elastic Net, many of the features are

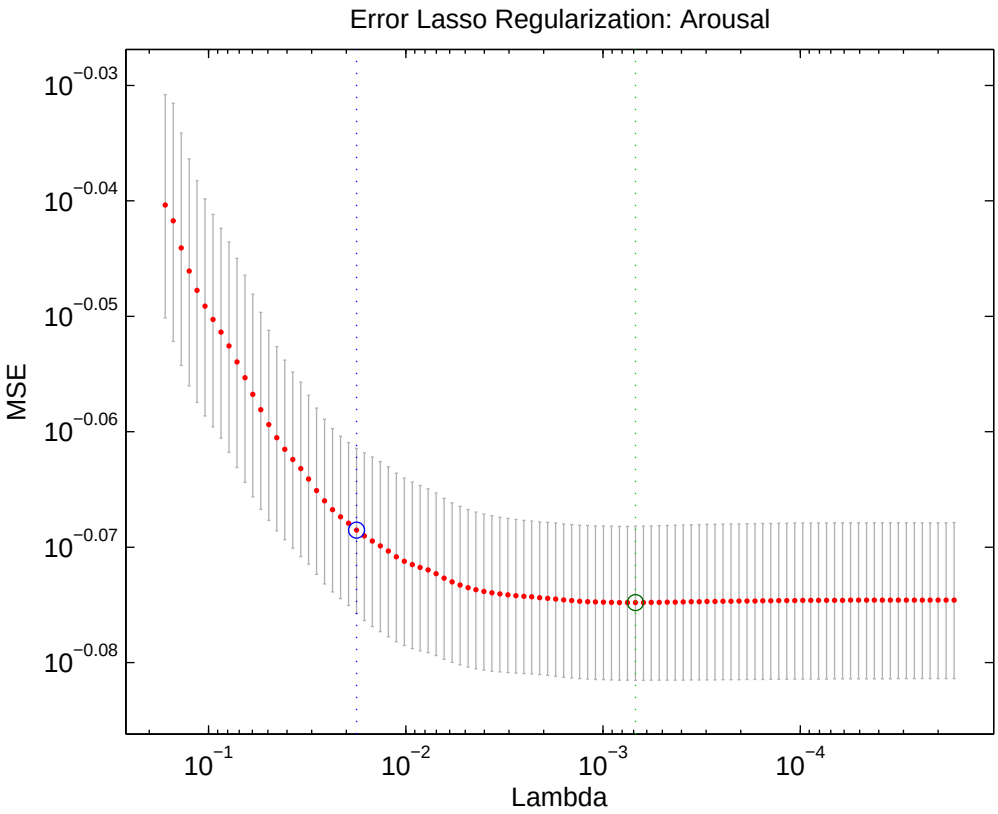


Figure 4.9: RMS for the different lasso free parameters, green lines indicate values that have the smallest errors; blue lines indicate values with the most zero parameters.

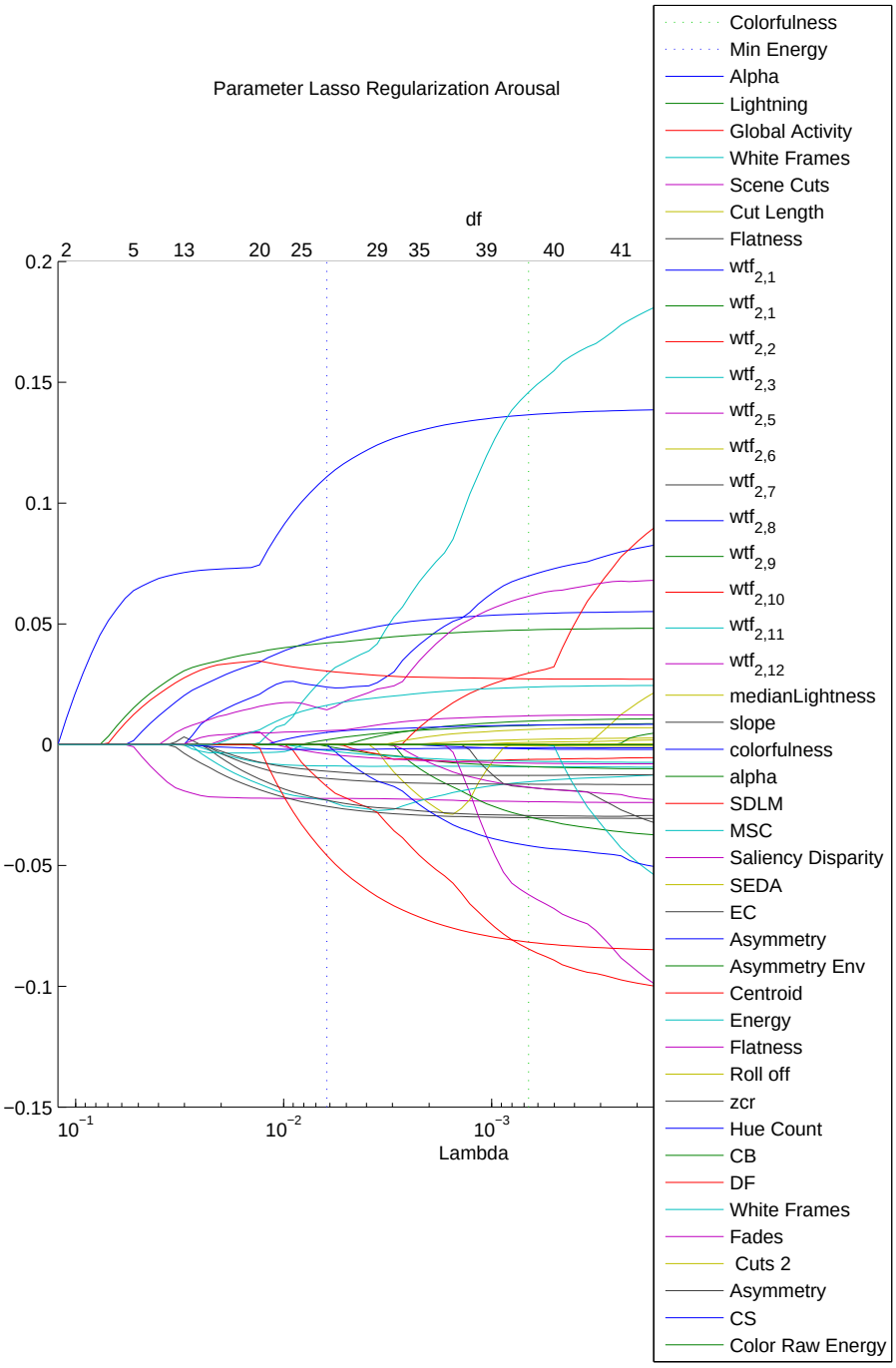


Figure 4.10: Trace plot of all the regression coefficients: green lines indicate values that have the smallest errors; blue lines indicate values with the most zero parameters (all values larger then green line are redundant) .

correlated and therefore are useful when used together. Fig. 4.13 to Fig. 4.14 show that valence values seem to be equally influenced by arousal and valence features. Overall most of the features when combined have some predictive power.

Bias-Variance

The bias variance decomposition for ridge regression with respect to the valence model is shown in Fig. 4.16, with a boot strap data set used 100 times. As expected the variance decreases as the regularization term increases. The regularization value seems to give good performance for variance until its natural logarithm reaches two. Similarly the bias suddenly increases at four. The bias variance decomposition for ridge regression with respect to the arousal model is shown in Fig. 4.16. The regularization value seems to give good performance until its natural logarithm reaches three, for both bias and variance, a small regularization value of 0.1 is all that is needed to stabilize the design matrix.

4.4.2 Clustering Results

Toy Data

In this section, we explore the toy data as it better illustrates how the method works. Fig. 4.17 shows toy data available with the Matlab implementation at [115], with colours corresponding to class membership. Each time series is a line with some random noise and the class membership is determined by the slope. The method will demonstrate how the different kernels can improve classification. Examining the toy data in Fig. 4.17, it is evident that there is a positive correlation between the time series in the same clusters and a negative correlation between time series in opposite clusters.

Examining the results using the linear kernel in Fig. 4.18, we see that the linear kernel clusters all the time series correctly. Comparing the results with the RBF kernel in Fig. 4.19, we see that many of the time series in the center are incorrectly classified.

Using the non-linear toy data shown in Fig. 4.20, we see that the data class is determined by the radial distance of the data and an offset. Examining the results using the linear kernel in Fig. 4.21

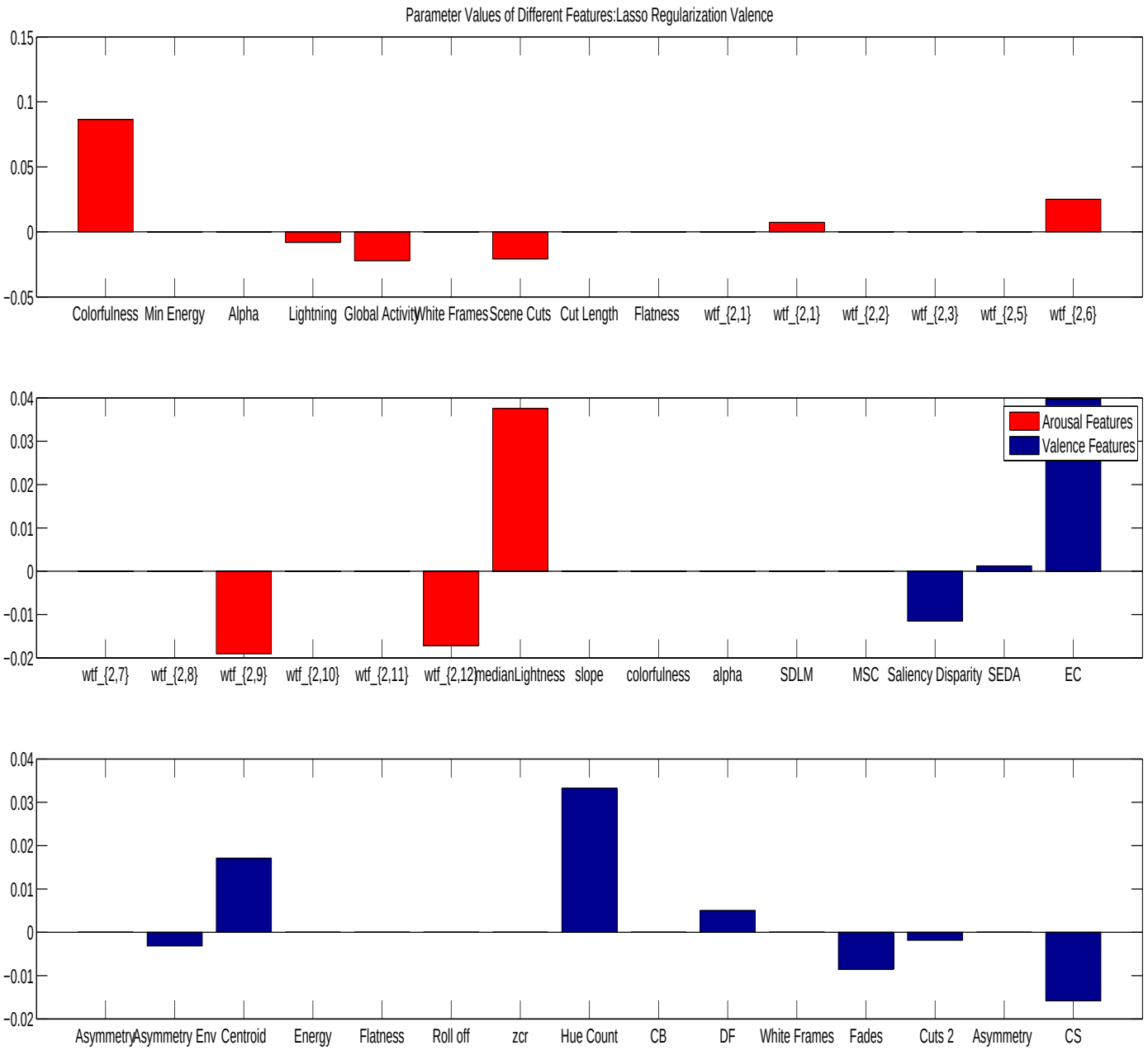


Figure 4.11: Optimum values of coefficients using Lasso and cross validation for arousal.

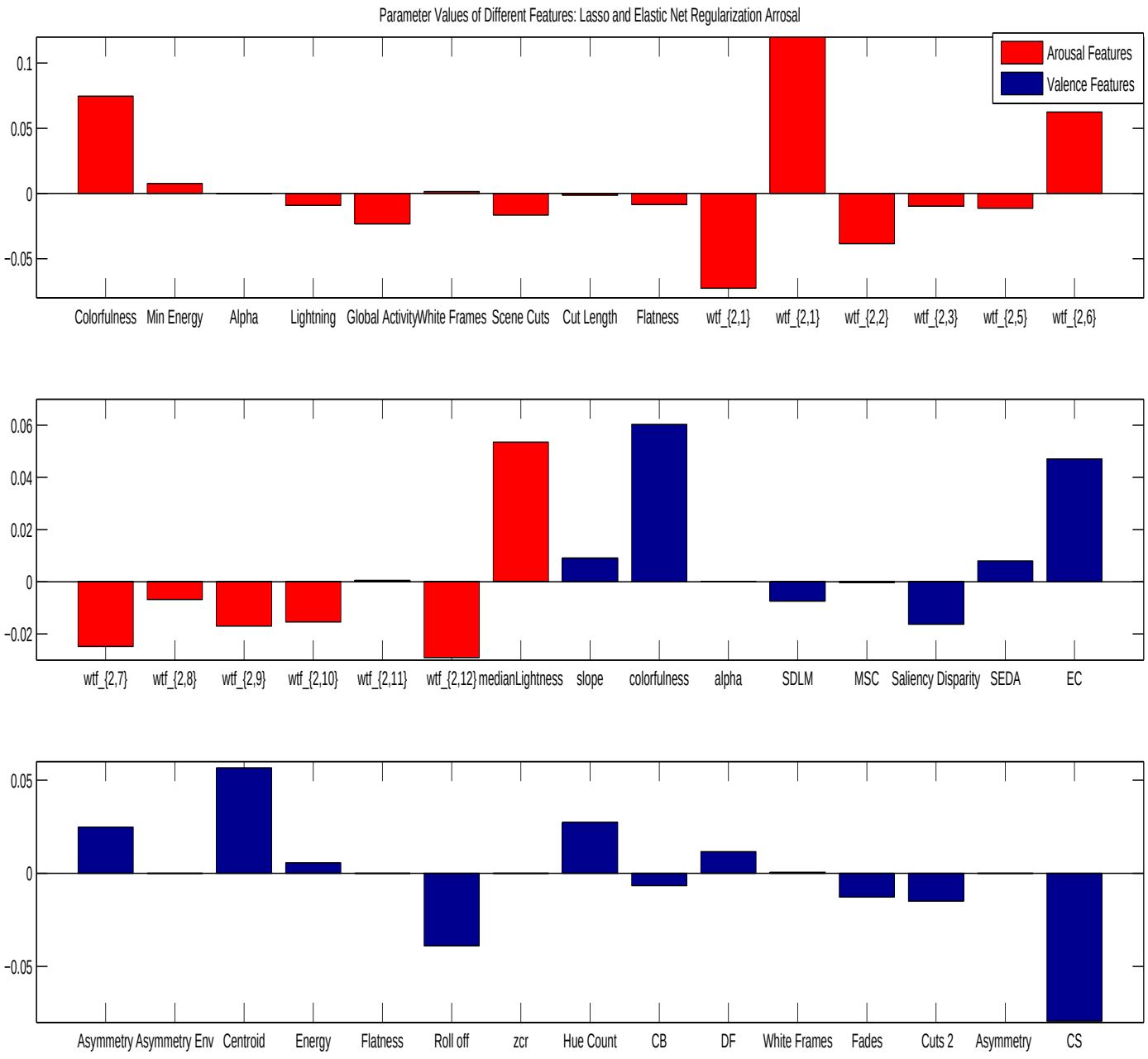


Figure 4.12: Optimum values of coefficients using Elastic Net and cross validation for arousal.

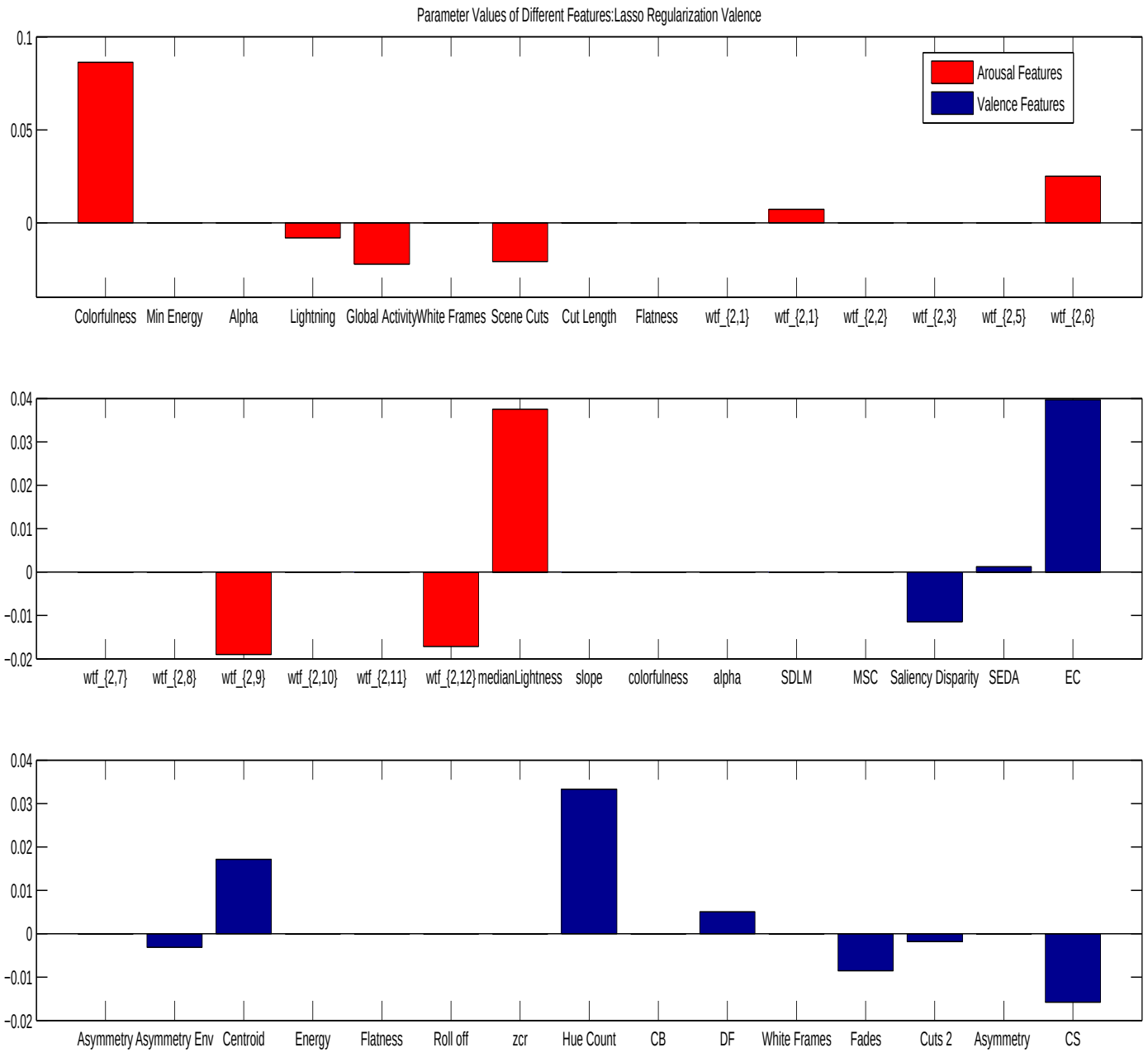


Figure 4.13: Optimum values of coefficients using Lasso and cross validation for valence.

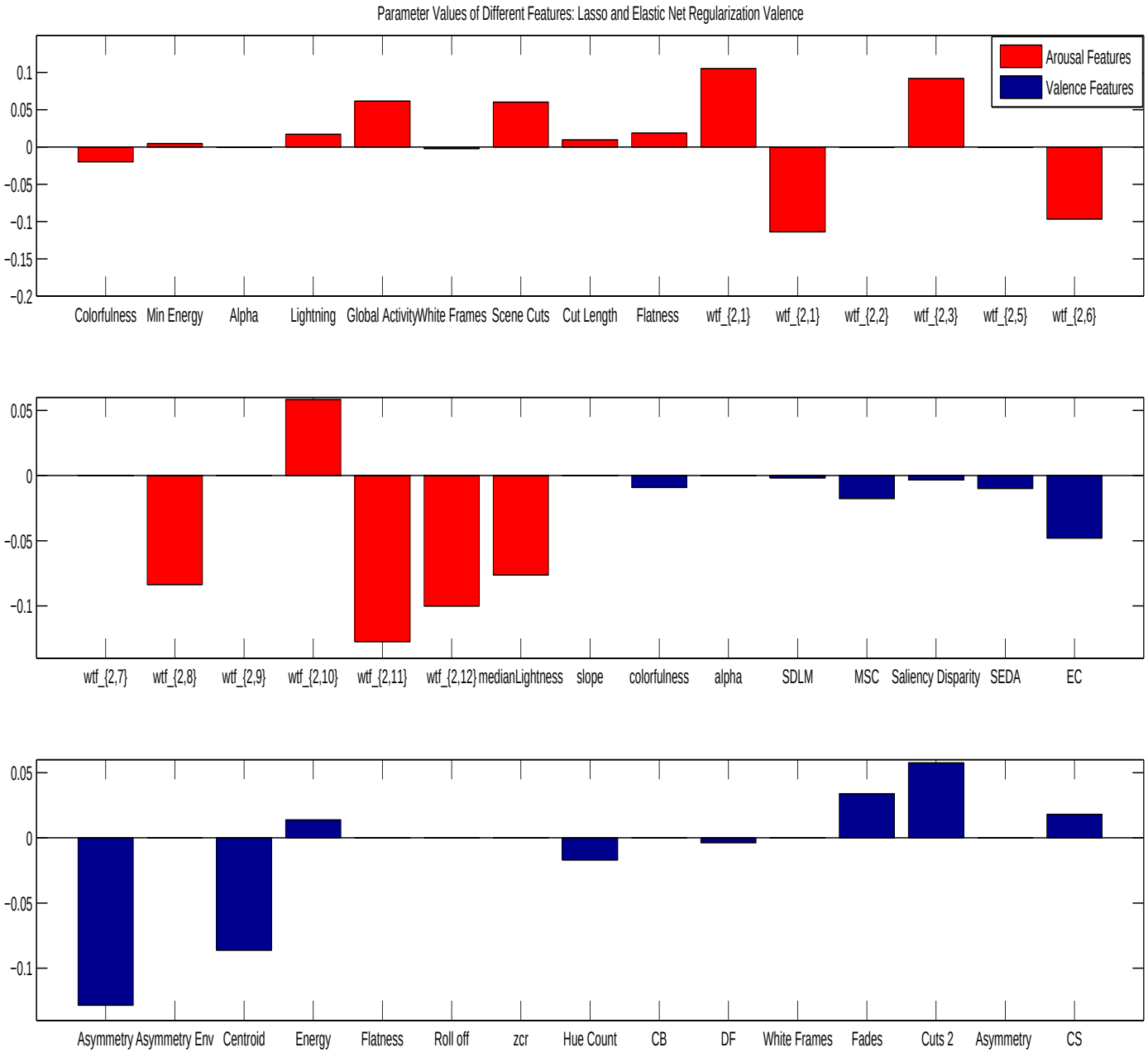


Figure 4.14: Optimum values of coefficients using Elastic Net and cross validation for valence.

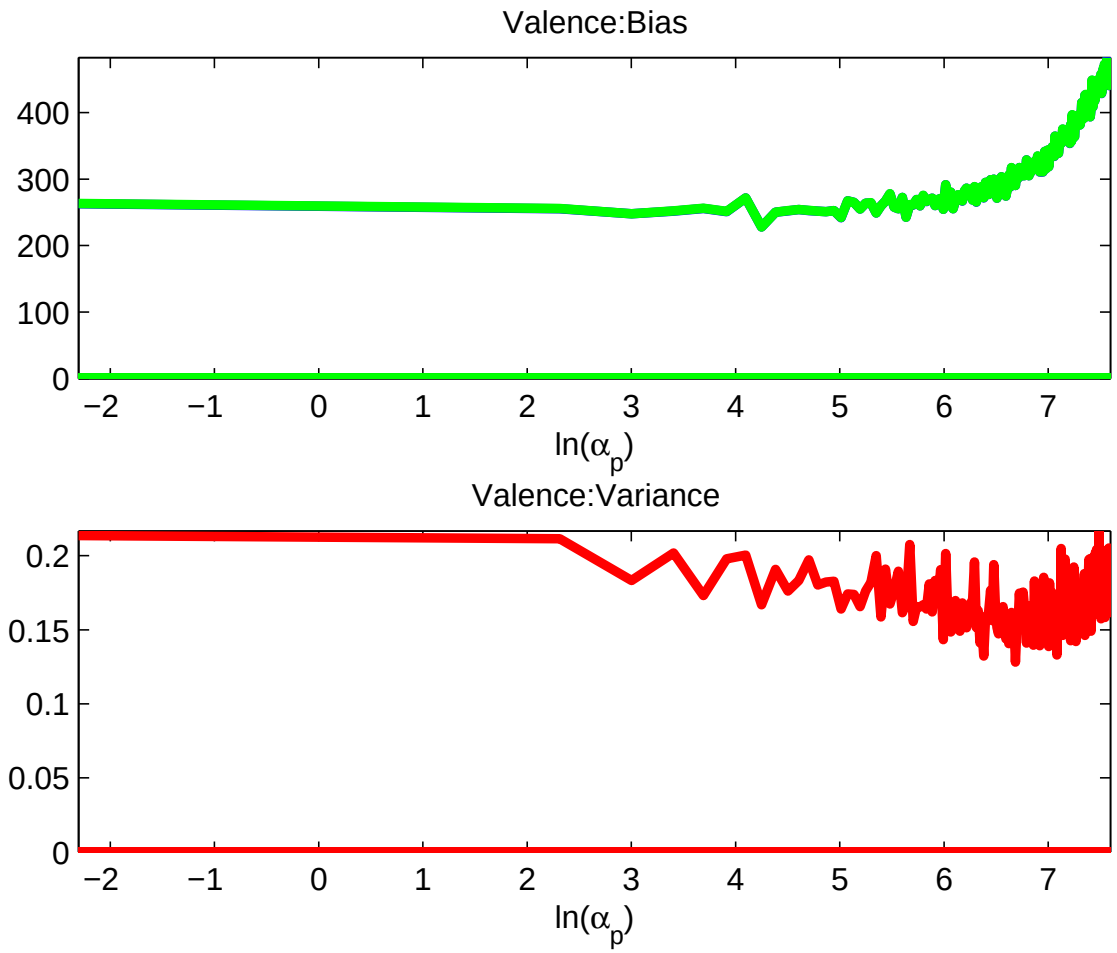


Figure 4.15: Top: Bias decomposition for different regularization parameter using valence model. Bottom: Variance decomposition for valence values over regularization parameter using valence model.

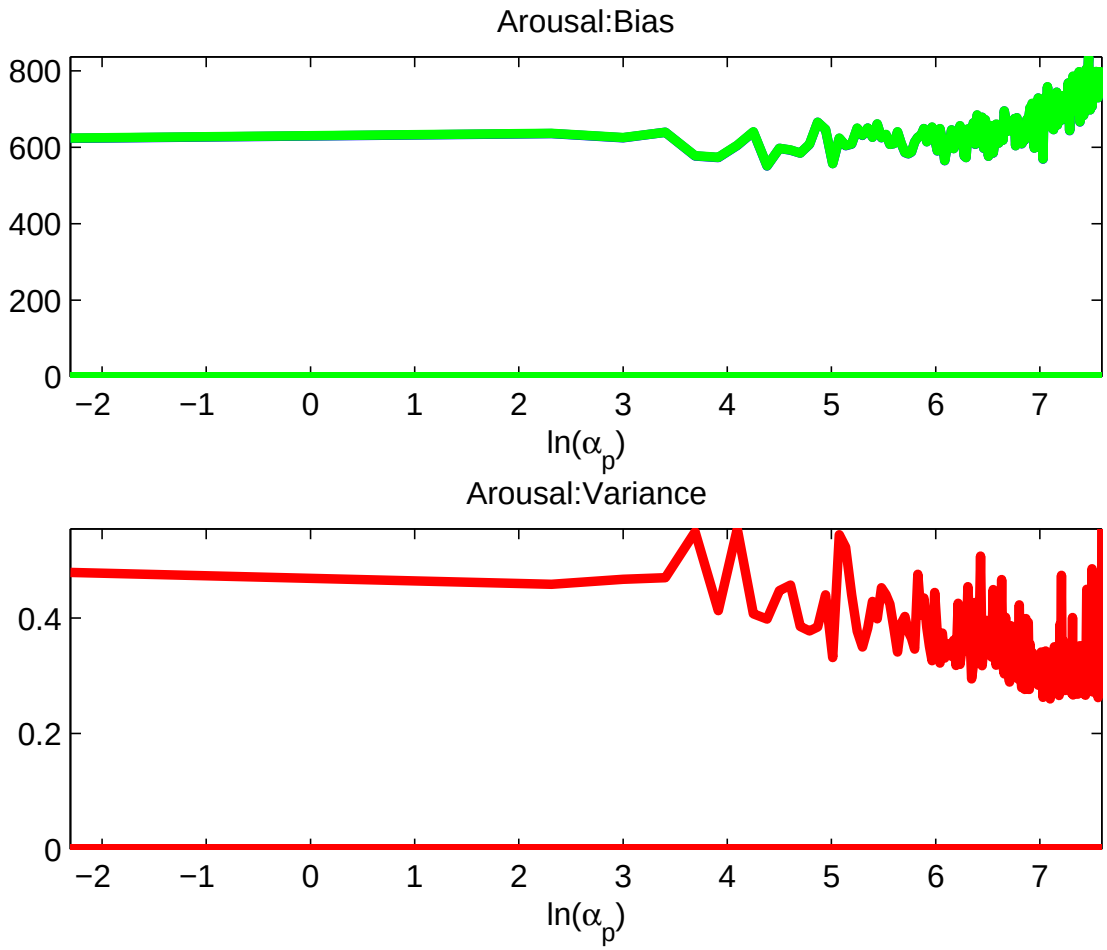


Figure 4.16: Top: Bias decomposition for different regularization parameter using arousal model. Bottom: variance decomposition for valence values over regularization parameter using arousal model .

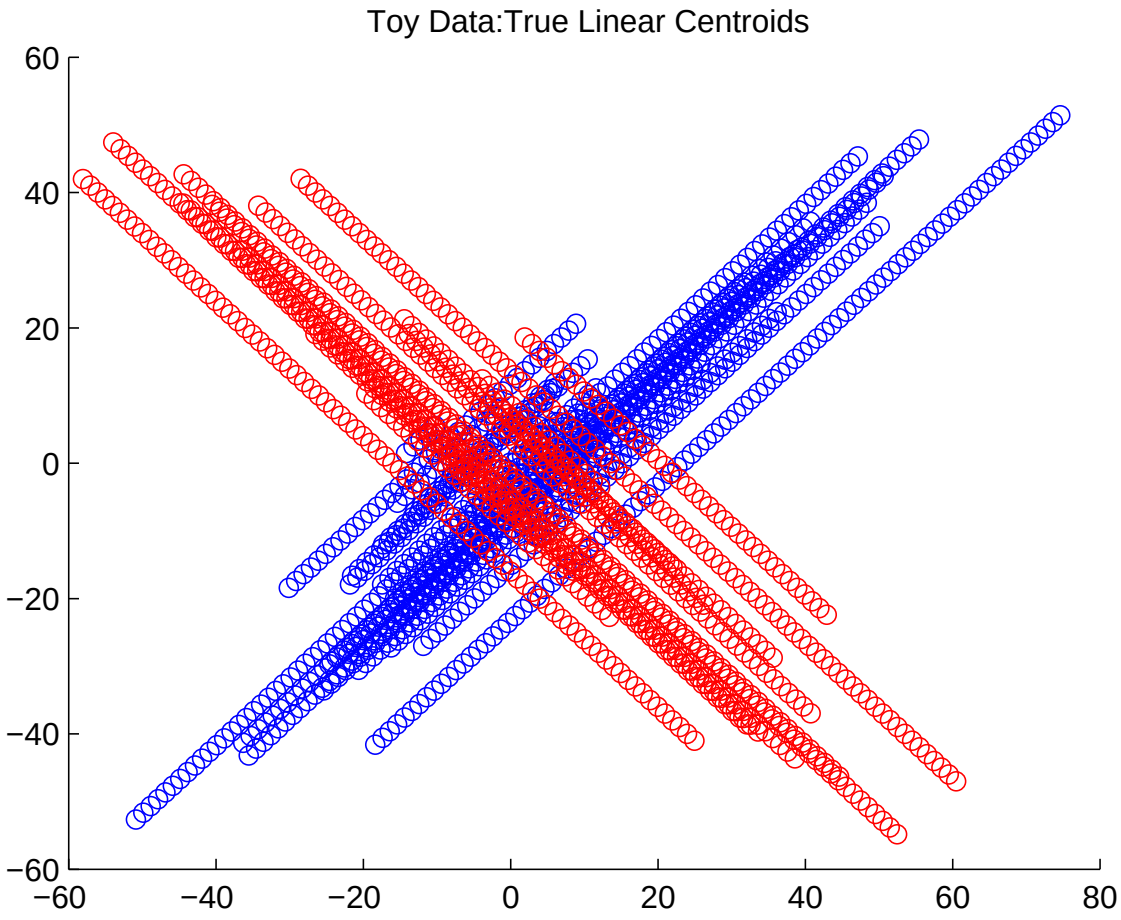


Figure 4.17: Linear toy data with color corresponding to labels.

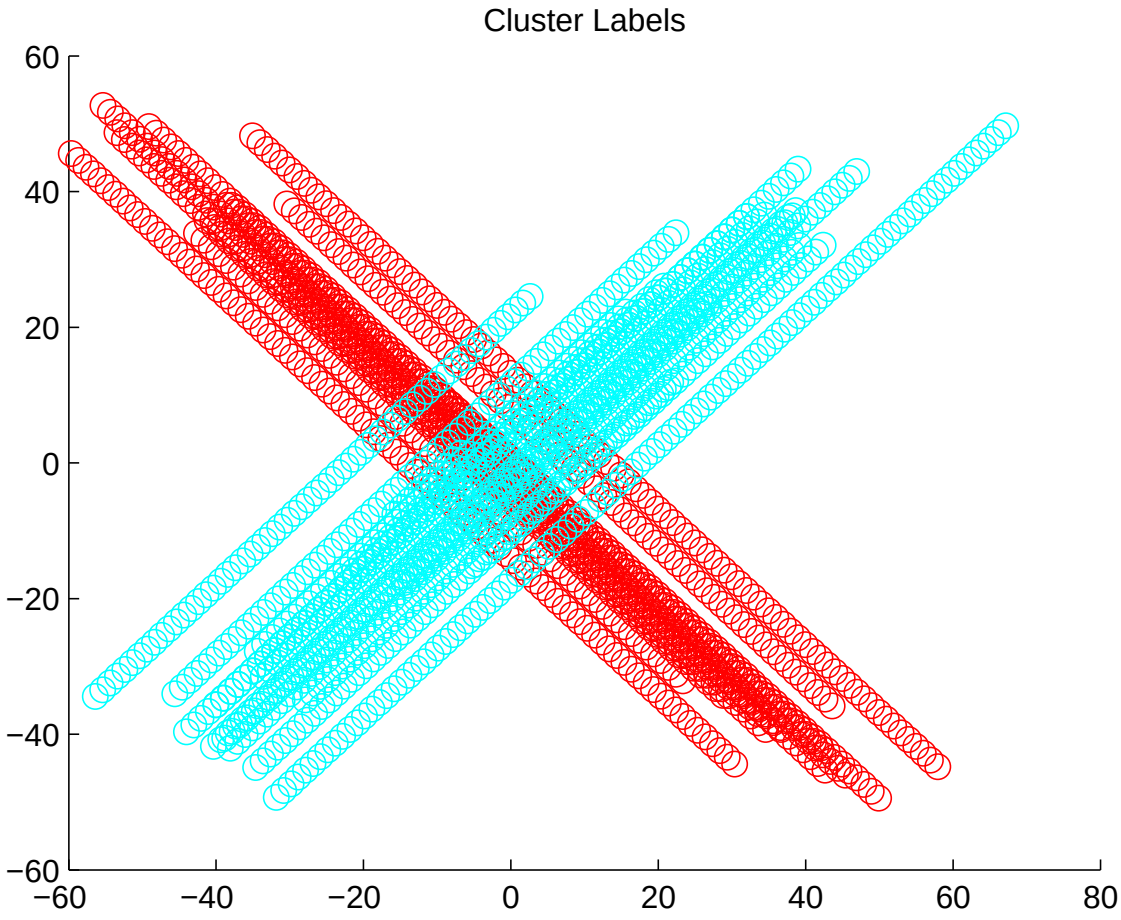


Figure 4.18: Results of using linear kernel on linear toy data with color corresponding to cluster labels.

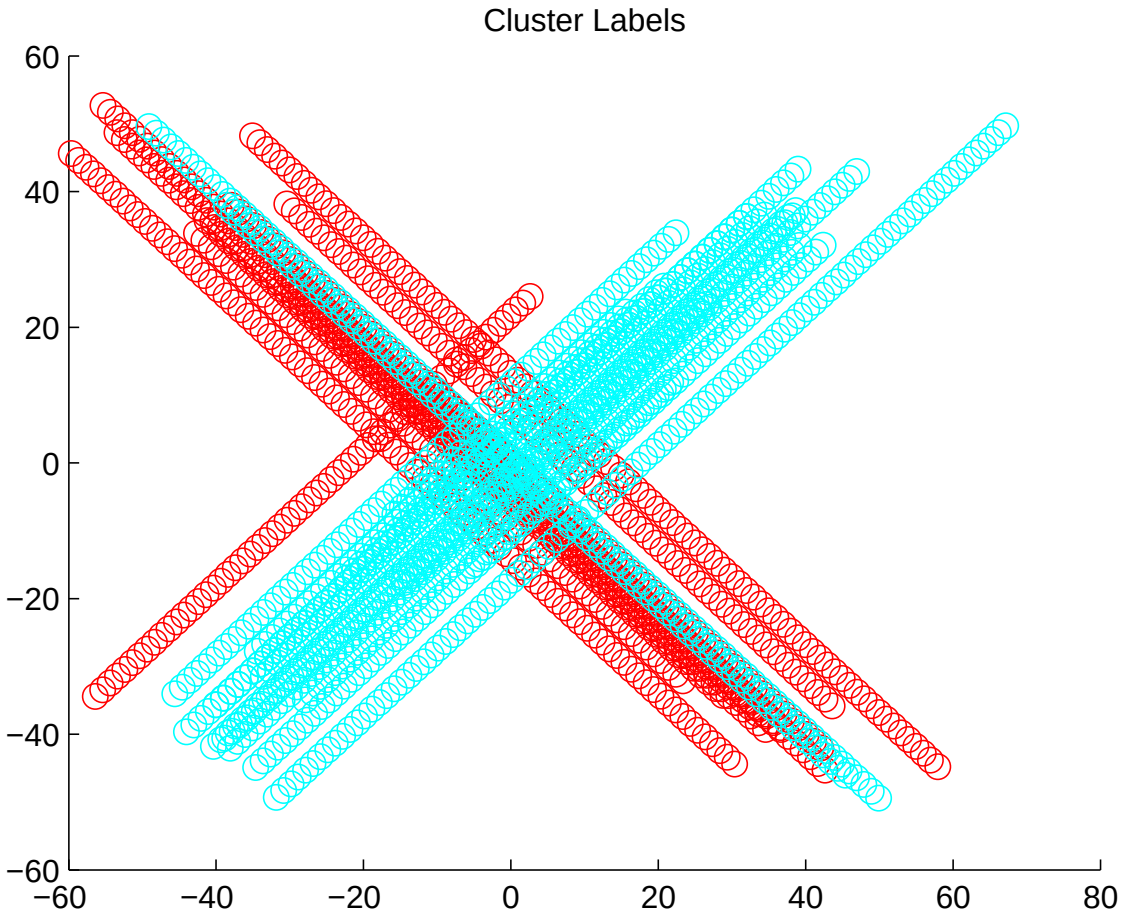


Figure 4.19: Results of using RBF kernel on linear toy data with color corresponding to cluster labels.

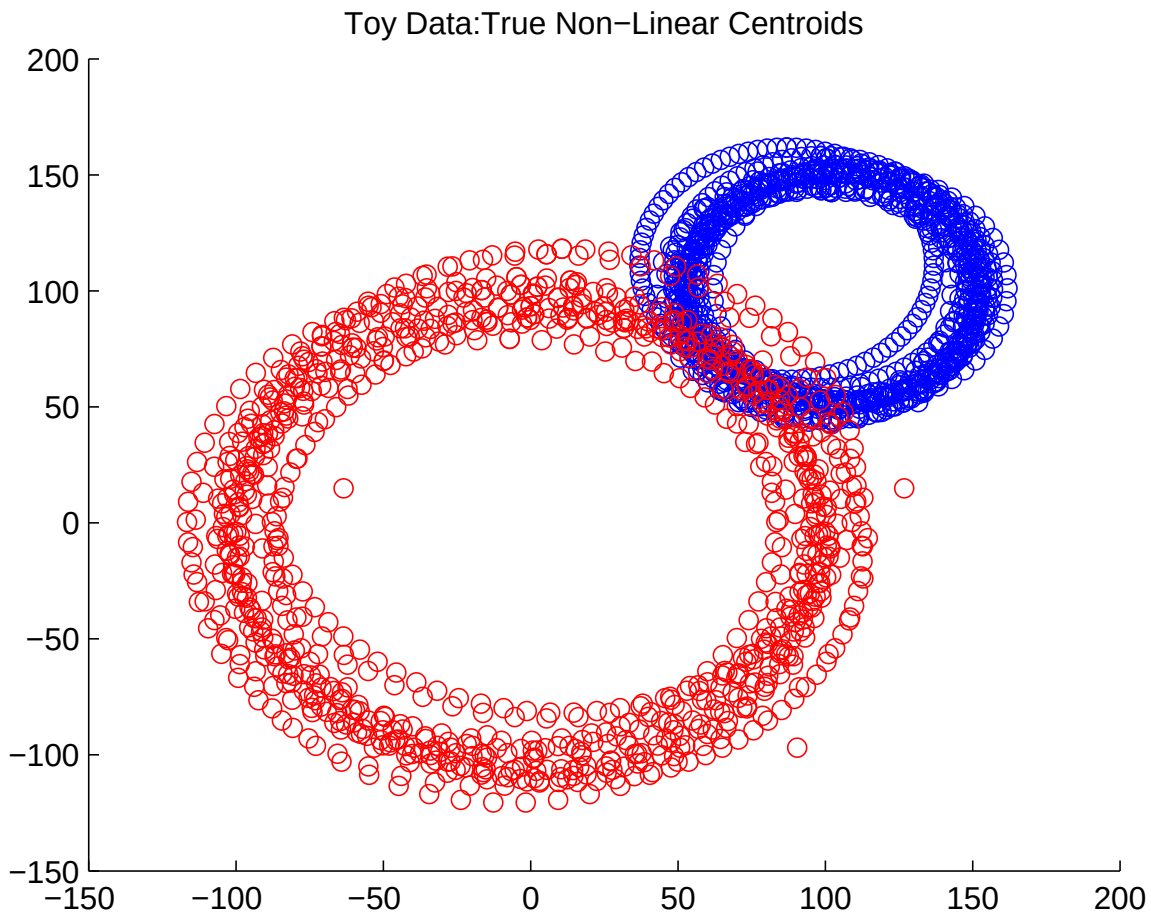


Figure 4.20: Non-Linear toy data with color corresponding to labels.

clusters everything into incorrect clusters. Comparing the results with the RBF kernel in Fig. 4.22, we see that all the time series are correctly classified. This demonstrates how different kernels can deal with different cluster time series geometry.

Real Data

In this section, we compare the novel clustering method to DTK and DNM. The experiments have been performed 100 times and averaged to avoid disparity due to cluster initialization. The results for average accuracy are included but as they are less informative we do not discuss them.

Table. 4.4 and 4.5 shows the average accuracy per sequence and accuracy for ranking valence.

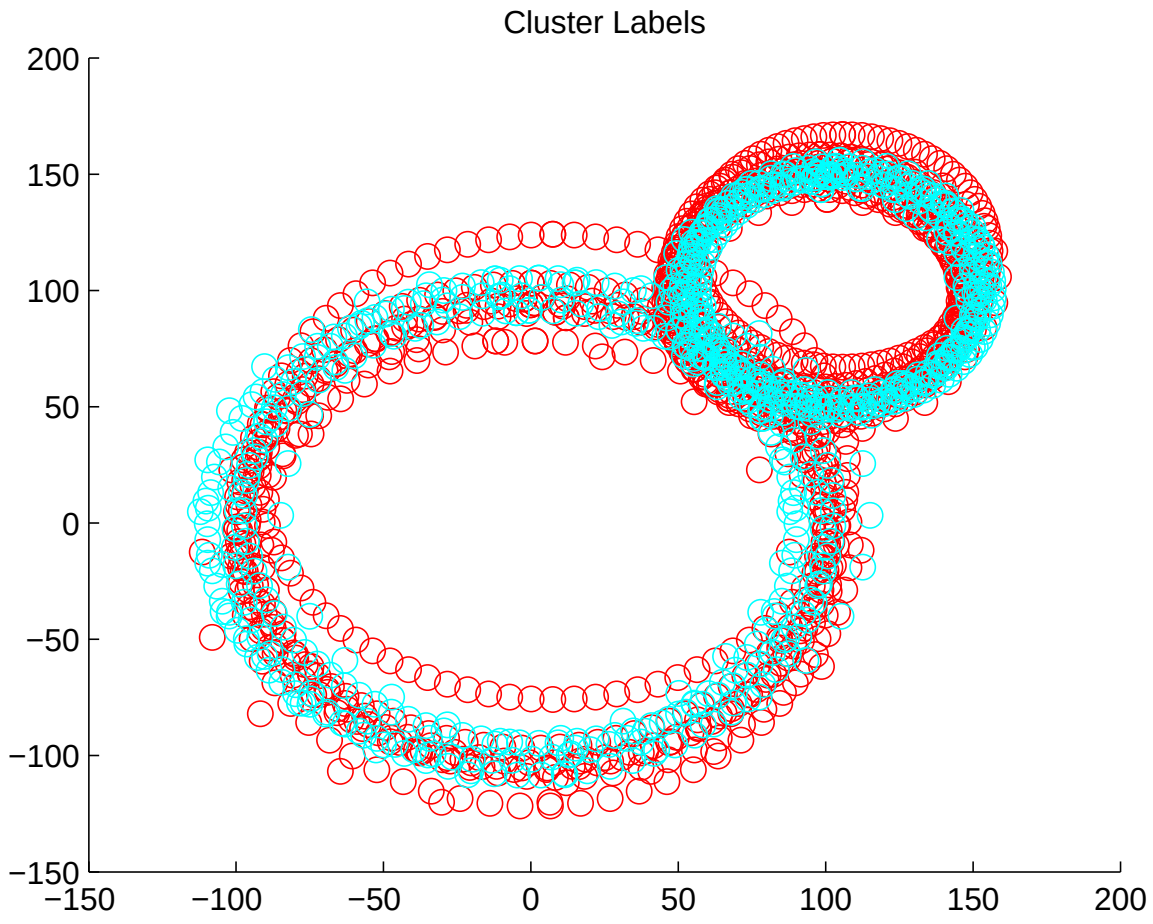


Figure 4.21: Results of using linear kernel on non-linear toy data with color corresponding to cluster labels, plus a bias.

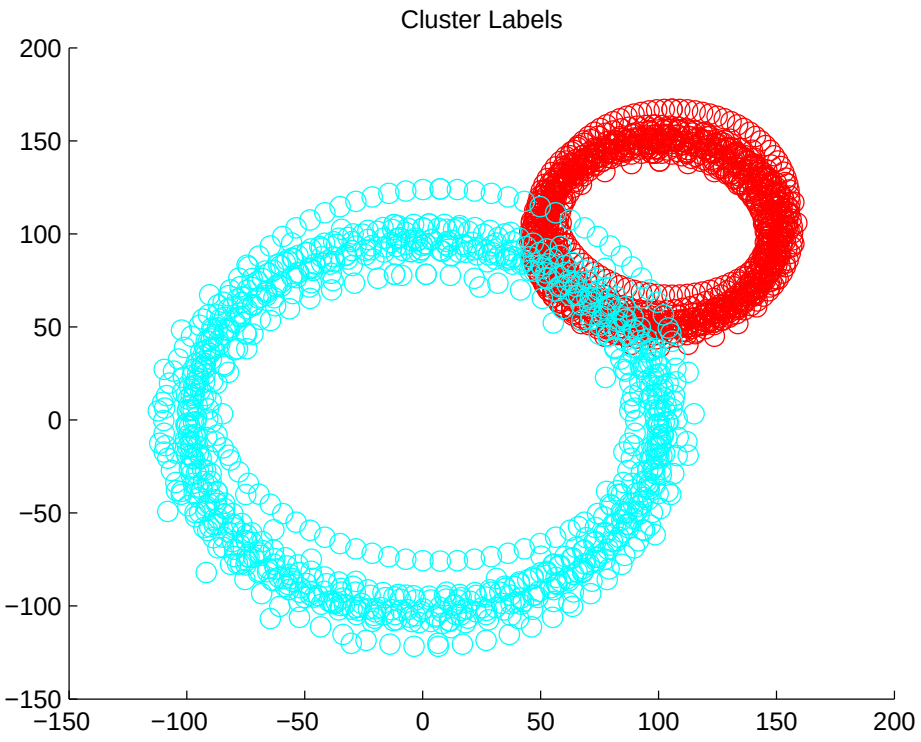


Figure 4.22: Results of using RBF kernel on linear toy data with color corresponding to cluster labels.

The RBF kernel performs best with an average accuracy per sequence of 79% and for three and four clusters the average accuracy per sequence is 58% and 49%. It appears that for a larger number of clusters the type of kernel does not play a role. The RBF works better as it maps the feature space to an infinite dimension. The scaling factor (SF) also plays a role in improving accuracy. It is observed that values over 100 do not improve performance.

Table 4.4: Average accuracy per sequence ranking valence using valence values.

Cluster	Linear	Quadratic	RBF (SF=10)	RBF (SF=100)
2	(76.26%,2.01%)	(76.14%,2.00)	(75.74%,2.01)	(78.54%,0.0201)
3	(60.88%,2.37)	(58.88%,1.92)	(58.88%,1.91)	(58.40%,2.16)
4	(49%,1.95)	(48.88%,1.78)	$(49.88\%, 1.78)$	$(48.88\%, 2.28)$

Table 4.5: Average accuracy for ranking valence.

Cluster	Linear	Quadratic	RBF (SF=10)	RBF (SF=100)
2	(63.88%,0.01)	(62.00%,0.01)	(63.48%,0.01)	(64.88%,0.01)
3	(40%,2.37)	(36.80%,4.38)	(43.17%,1.07)	(42.40%,2.16)
4	(30%,3.7)	(30.88%,9.98)	(31.12%,2.70)	(31.21%,3.7)

Table. 4.6 and 4.7 show the average accuracy and average accuracy per sequence for ranking in the arousal axis. For arousal, the type of clusters does not seem to play a role. For two clusters, the average accuracy per sequence and accuracy is approximately 75%. For four clusters the results are 44%. For three clusters, RBF kernels have slightly better accuracy with an average accuracy per sequence of 56%, with other kernels having an accuracy of 54%.

Table 4.6: Average Accuracy per Sequence of unsupervised method for ranking on the arousal using the arousal axis.

Cluster	Linear	Quadratic	RBF (SF=10)	RBF (SF=100)
2	(75.63%,1.79)	(75.12%,1.70)	(75.53%,1.79)	(75.53,0.0201)
3	(54.88%,1.77)	(54.88%,1.88)	(54.54%,1.91)	(**56.40%**,2.16)
4	(43.88%,1.00)	(43.88%,1.78)	(43.54%,2.36)	(41.21%,3.7)

The classification results using the DTK are given in Table. 4.8. For two clusters and three clusters the results for arousal are similar. For two clusters the average accuracy per sequence

Table 4.7: Average accuracy of unsupervised method for ranking arousal.

Cluster	Linear	Quadratic	RBF (SF=10)	RBF (SF=100)
2	(53.34%,1.79)	(53.54%,3.63)	(53.54%,1.51)	(53.54%,1.41)
3	(41.72%,2.97)	(41.88%,0.80)	(41.88%,1.71)	(43.53%,1.78)
4	(32%,1.95)	(32.10%,1.80)	(32.22%,1.50)	(31.2%,1.89)

is approximately 75% and for three clusters the average accuracy per sequence is approximately 54%. This is almost identical to DTAKKC. When comparing results for four clusters the average accuracy and average accuracy per sequence are about 10% less. WHM performed relatively worse on valence. All the values are 4% to 8% worse for different cluster values.

Table 4.8: Average Accuracy (AA) and Average Accuracy per Sequence (AAS) of DTK using different clusters

Cluster	AA DTK Valence	AAS DTK Valence	AA DTK Arousal	AAS DTK Arousal
2	(50.13%,7.79)	(**75.12%**,2.70)	(50.46%,1.79)	(**75.53%**,1.79)
3	(40.88%,0.80)	(50.88%,1.91)	(33.15%,4.56)	(54.54%,1.17)
4	(25.88%,1.00)	(46.88%,1.78)	(25.04%,2.65)	(31.21%,1.58)

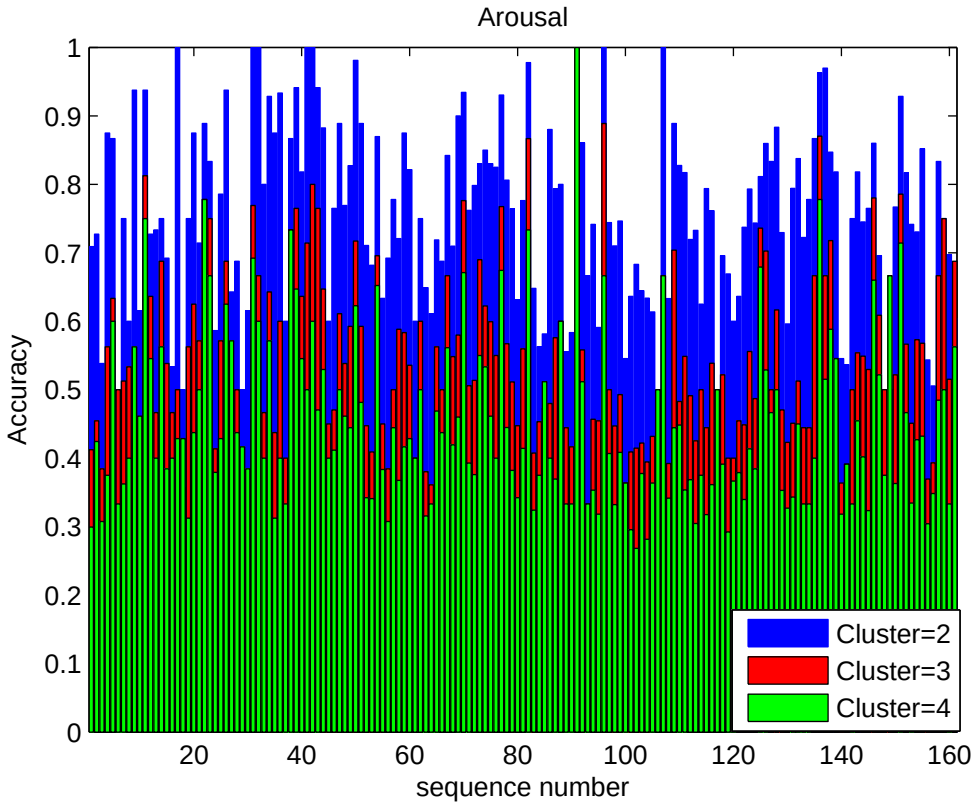
The results for WHM are shown in Table. 4.9. While the method performs comparably to DTAKKC using arousal, the method performs poorly using valence.

Table 4.9: Average Accuracy (AA) and Average Accuracy per Sequence (AAS) of WHM using different clusters.

Cluster	AA Arousal	AAS Arousal	AA WHM Valence	AAS Valence
2	(50.44%,8.50)	(75.54%,2.01)	(50.07%,1.79)	(50.07%,3.73)
3	(34.30%, 4.46)	(54.54%,1.74)	(33.15%,4.56)	(33.27%,0.01)
4	(25.05%,2.68)	(45.88%,1.58)	(25.04%,2.65)	(25.21%,0.01)

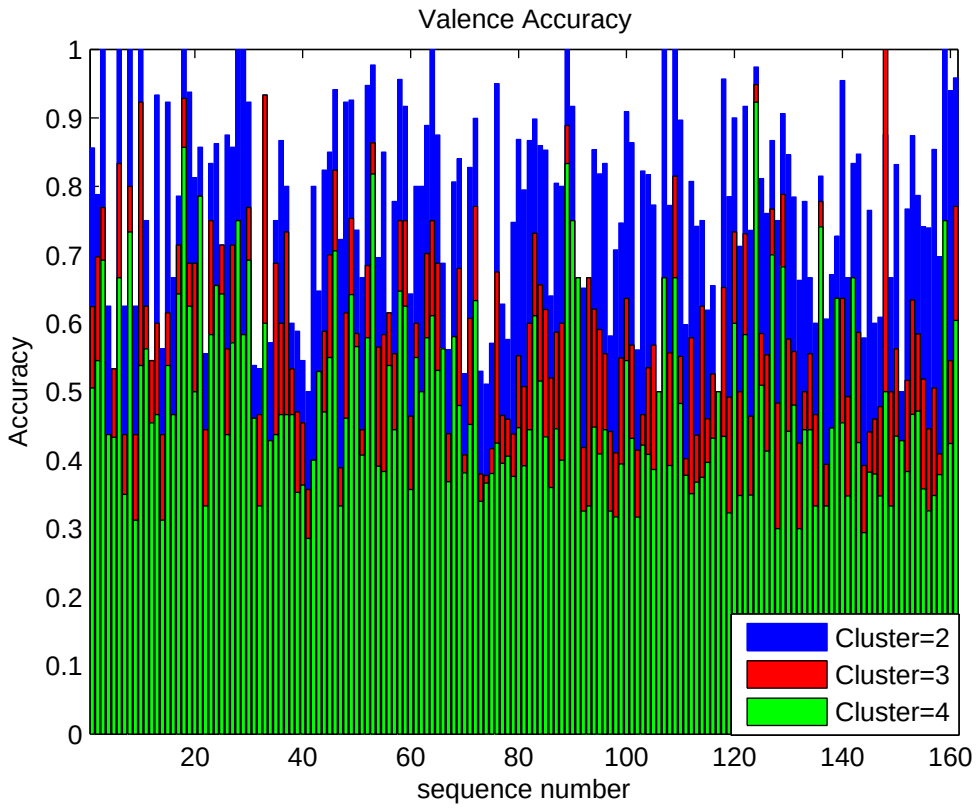
The accuracy of each sequence using arousal and valence is shown in Fig. 4.23 and Fig. 4.24 respectively. The order on the bottom axis is the same as that given in the LIRIS database. For this example the linear kernels are used in each case.

Intuitively, longer sequences should have less accuracy as there is a wider range of emotions. This is demonstrated in Fig. 4.25, where the accuracy is plotted versus the length of the sequence,



,width=12 cm

Figure 4.23: Accuracy of each sequence using arousal and linear kernels for different clusters.



,,width=12 cm

Figure 4.24: Accuracy of each sequence using valence and linear kernels for different clusters.

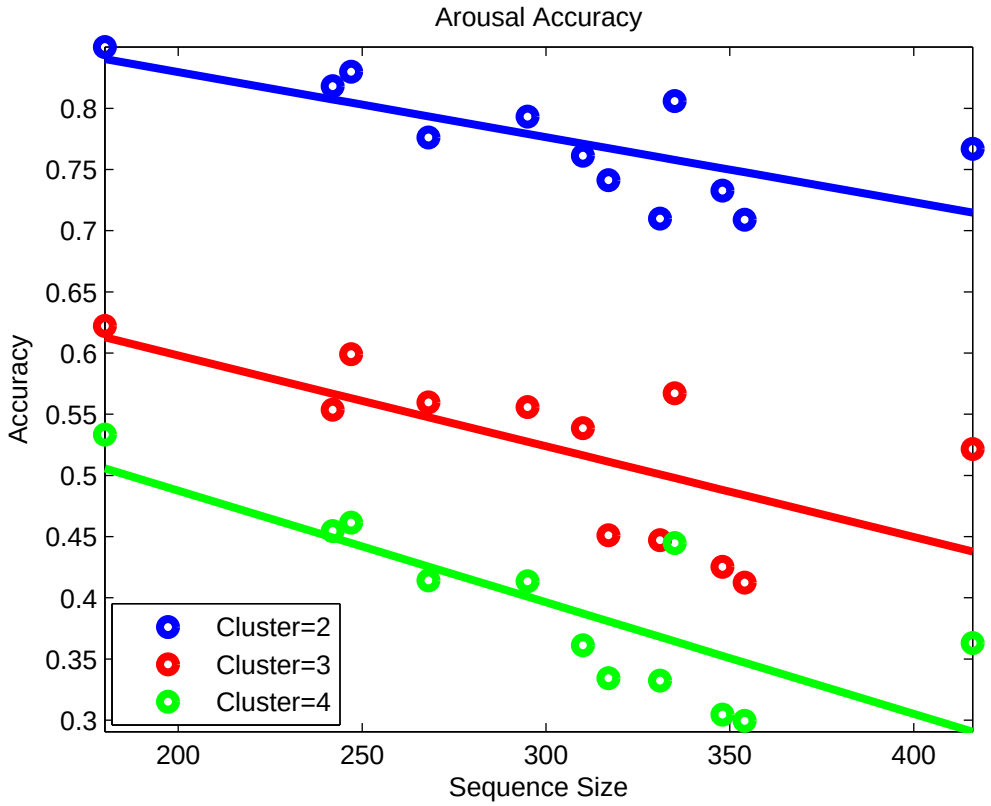


Figure 4.25: Accuracy plotted vs the length of the sequence with different colors repressing different cluster numbers.

with different colours representing different clusters. It is evident that, as the sequence length increases the accuracy decreases. Table. 4.10 shows different coefficients of determination for different cluster numbers. The length of the sequence is the independent variable and the accuracy is the dependent variable.; The values are calculated for different cluster numbers. It is evident that as the number of clusters increases, the length is a better predictor of the accuracy. An interesting observation is that the shortest sequences are more difficult to predict using sequence length. This can be seen by comparing the different rows of Table. 4.10, as the shorter sequences are trimmed, the coefficient of determination increases.

Using visual inspection, the results are much more apparent. In each figure the small circles represent a different video clip with its membership denoted by its colour. Each cluster's membership is determined using the entire sequence, hence the overlap. Fig. 4.26 and Fig. 4.27 compare

Table 4.10: R^2 Coefficient of determination for prediction average sequence accuracy for different lengths (cluster=2,cluster=3,cluster=4).

Lengths	Arousal	Valence
All Lengths	(6.45e-05,0.0178, 0.0621)	(0.0035,0.0442, 0.0540)
Lengths>100 samples	(2.61e-05, 0.0502, 0.1006)	(0.0010, 0.0165, 0.0092)
Lengths>150 samples	(0.5123,0.4360,0.6207)	(0.0970, 0.1537,0.2319)

the novel clustering method to DTK using the valence value $\hat{\mathbf{y}}_m = \hat{y}_v(\mathbf{x}_{m,v})$. Fig. 4.26 illustrates the novel clustering methods. It is evident that the different clusters correspond to different levels of valence: red values correspond to high-valence, green are medium-high, blue are medium low, and purple are low. There is much less overlap compared to DTK shown in Fig. 4.27 where the clusters marked by green and blue totally overlap. In addition, the cluster marked by red corresponding to high valence appears to have a considerably large number of values on the low valence side.

Fig. 4.29 and Fig. 4.30 compare DTAKK to WHM respectively, and it is evident that WHM method has little correspondence to the valence values.

Fig. 4.32 demonstrates how the clusters for valence relate to conventional emotion recognition methods. Different colours represent regions that seem to be responsible for different cluster membership. The colour correspondence is the same as in Fig. 4.31, where series that are marked in red contain positive emotions such as happiness. Series in the green region have less positive emotions but are still positive and seem to have a wider range, from excited to tired. Series in blue contain series that consist of videos that are non-positive and also have a large range from alarmed to gloomy, and purple consist of series with emotions such as miserable and afraid.

Fig. 4.33 and Fig. 4.34 show the results using the arousal values, with three clusters for both methods: the novel method and DTK. Examining the novel method Fig. 4.33, we see that there is a clear relationship with red corresponding to sequences with high arousal, green corresponding to medium arousal and blue indicating features with low arousal. Examining DTK we see that there is no medium value for arousal as the clusters marked by blue and red totally overlap. Furthermore, red samples corresponding to red clusters totally encompass the other clusters.

Fig. 4.37 and Fig. 4.38 display the results using DTAKKC and DTK respectively. Table. 4.11

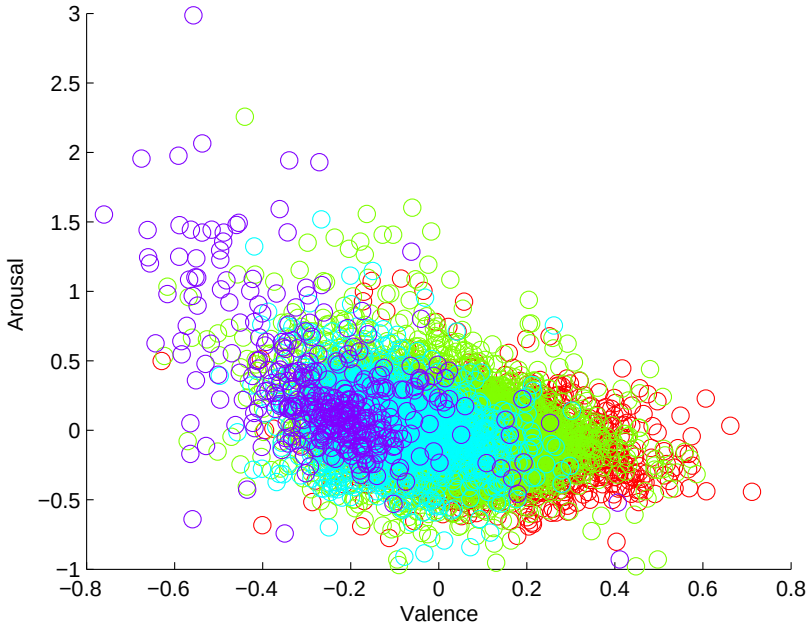


Figure 4.26: DTAKKC Linear Kernel

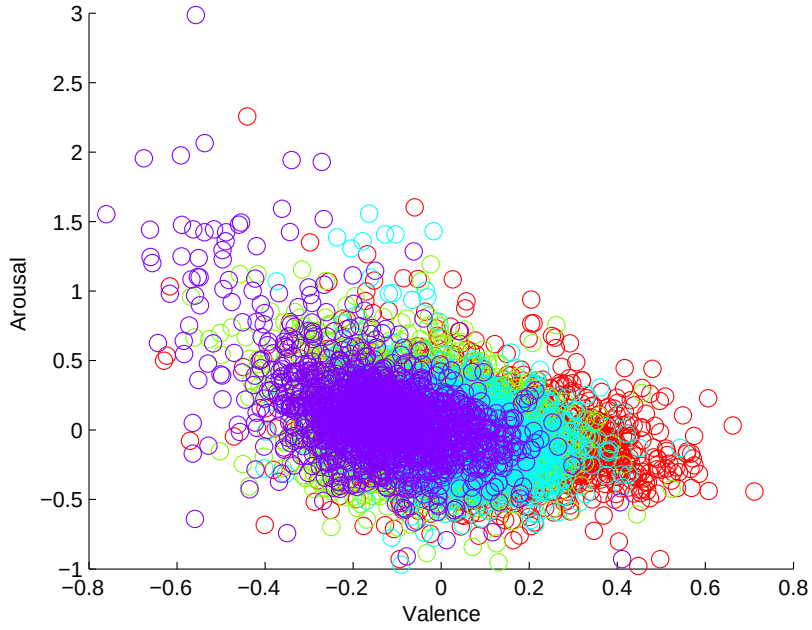


Figure 4.27: DTK

Figure 4.28: DTAKKC compared to DTK using 4 clusters performed on valence values.

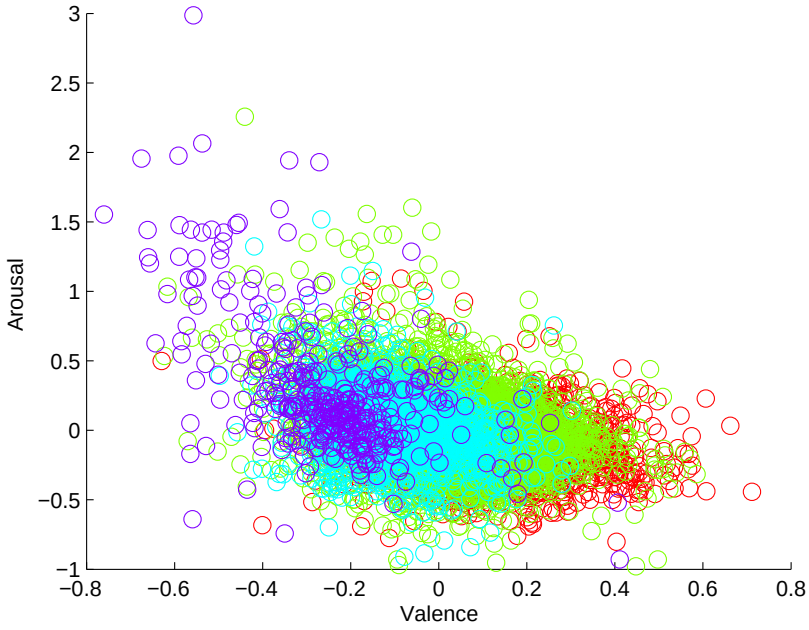


Figure 4.29: DTAKKC Linear Kernel

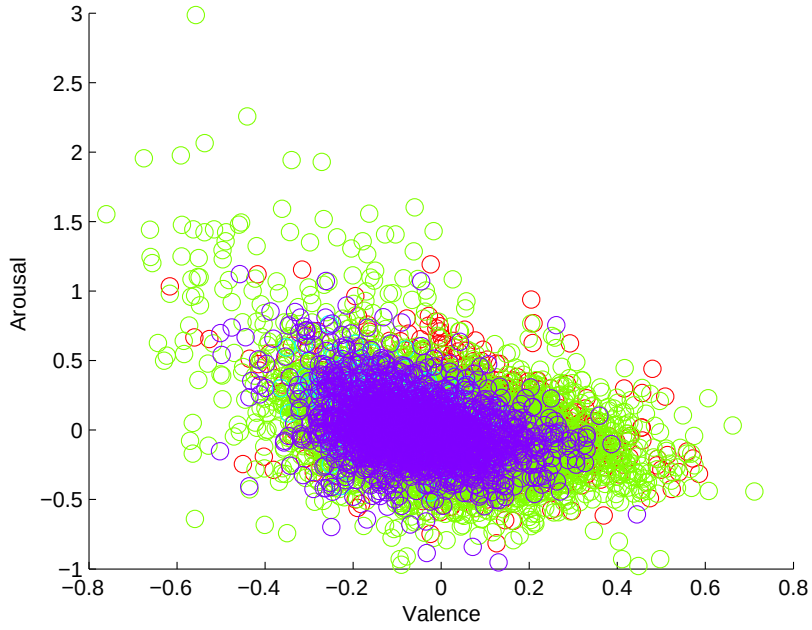


Figure 4.30: WHM

Figure 4.31: WHM compared to DTK using 4 clusters performed on valence values.

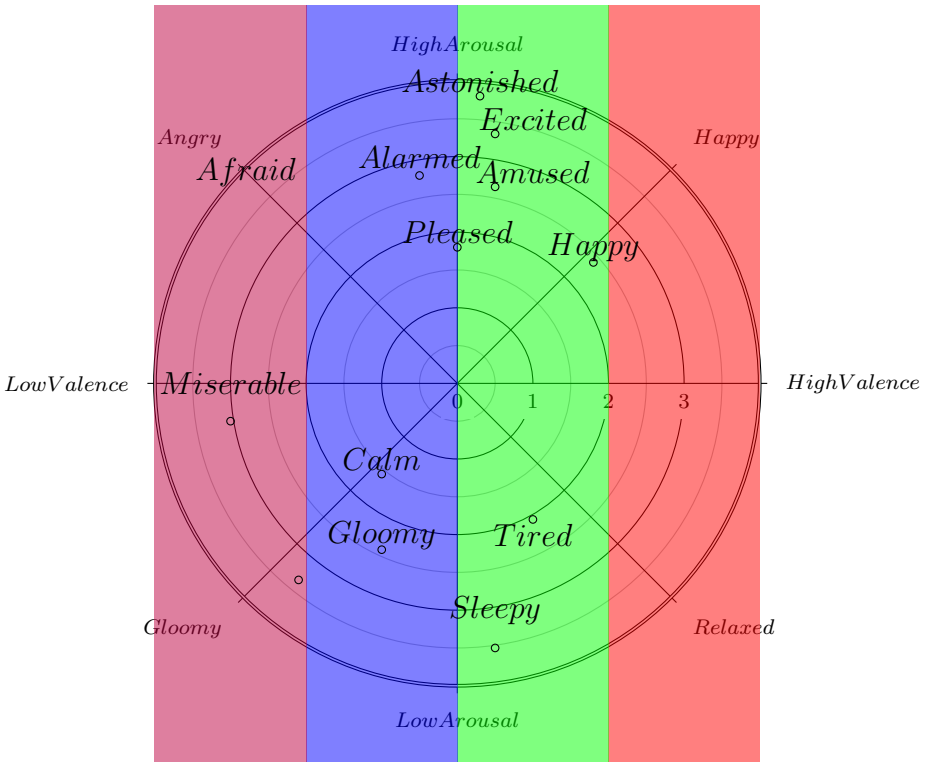


Figure 4.32: Different colours representing regions that correspond to different cluster memberships for valence.

gives some movie titles corresponding to the clusters in Fig. 4.37, with the indexes of the video clips that comprise the movie from the database. It is evident that DTK has no discernable pattern. Examining Fig. 4.37 and comparing to DTK we see the center purple cluster corresponds to neutral content. Titles from this cluster are dialogue heavy and have little camera motion. Some titles are given in Table. 4.11.

The blue cluster appears slightly shifted to the left, compared to Fig. 4.1. The content in this cluster is associated with tension and distress. This corresponds with several examples given in Table. 4.11; these titles are about kidnapping and mental illness.

Videos in the green cluster appear to have the greatest range of emotion and correspond to the more typically entertaining movies with complex plots. Most of the activity appears in the top quadrants corresponding to fear and distress to joy. The content appears to vary the most in the cluster as well. For example, examining the third row in Table. 4.11 we see love stories, two action stories and a fast-paced drama. We also place the most likely conventional emotion recognition in the first column: fear, anger, sadness, happiness, disgust and surprise. It should be noted that many of the stronger emotions such as fear are represented in both clusters.

The films in the cluster marked with red are contained in the top left quadrant and associated with neglect, fear, anger and tension. Most of the contents in this section consist of horror movies and creepy art house films. For example, examine the fourth row in Table. 4.11 The Room of Franz Kafka, which is about the author Franz Kafka, well known for his oppressive and nightmarish work.

Fig. 4.36 demonstrates how the clusters for arousal relate to conventional emotion recognition methods. Different colours represent regions that seem to be responsible for different cluster membership. The colour correspondence is the same as in Fig. 4.35, whereby series that are marked in red contain a range of emotions from exciting to scary. Series in the green region seem to have a range of emotions, but not extremely intense. The series in the blue cluster contain tiring or boring content.

Examining Fig. 4.40, we see several images extracted from the films in Table. 4.11. It is evident that the films on the left side contain low valence content with violent scenes, while the films on

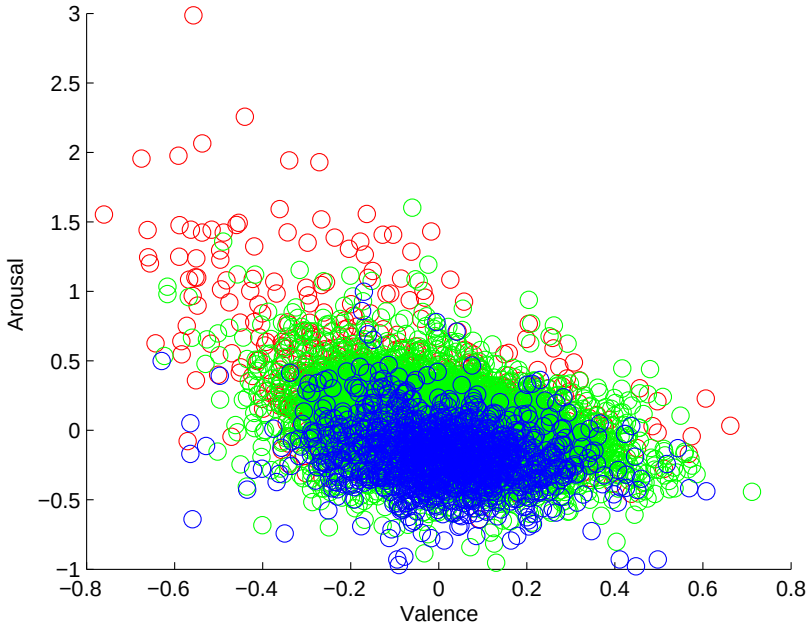


Figure 4.33: DTAKKC Linear Kernel

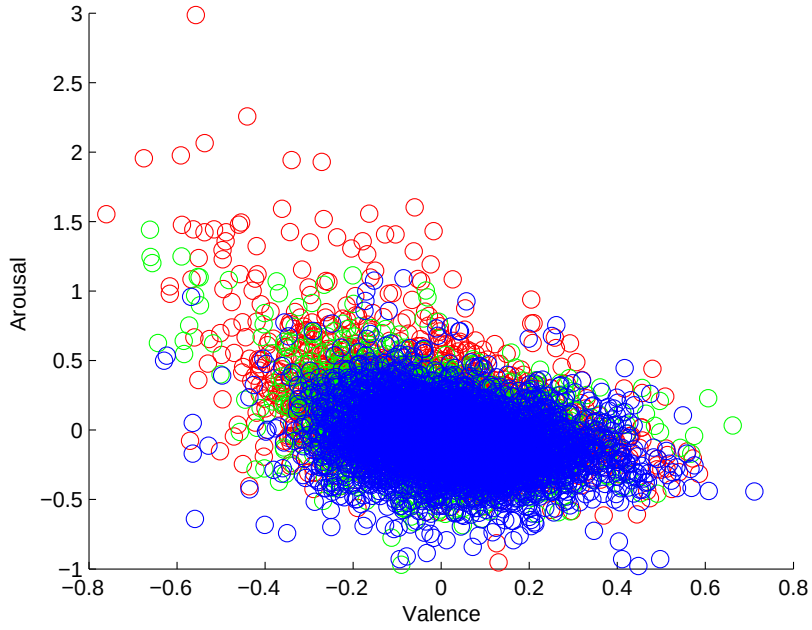


Figure 4.34: DTK

Figure 4.35: DTAKKC compared to DTK using 3 clusters performed on arousal values

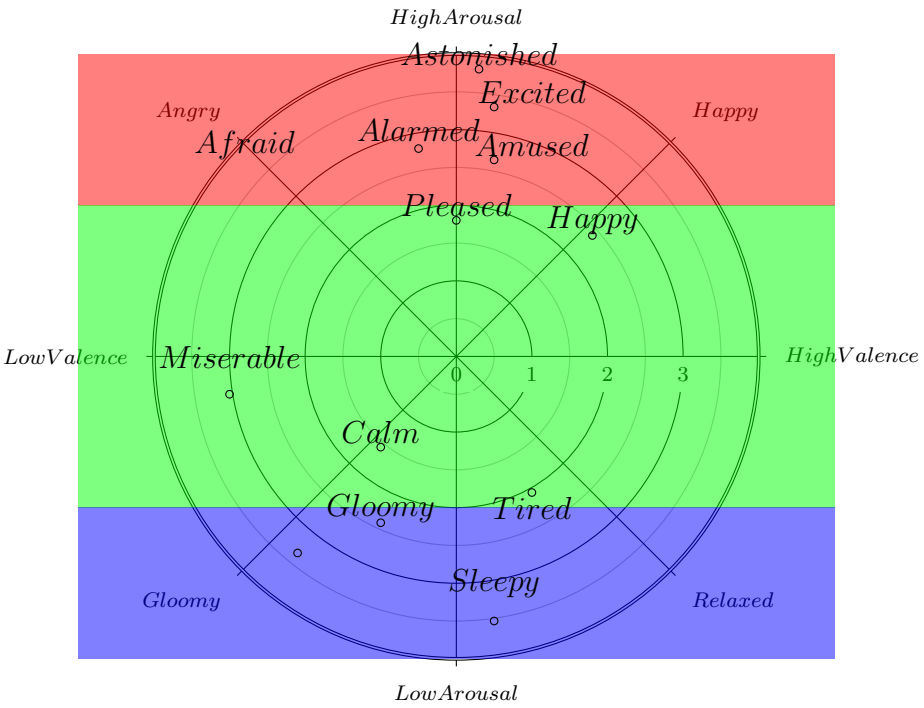


Figure 4.36: Different colors representing regions that correspond to different cluster memberships.

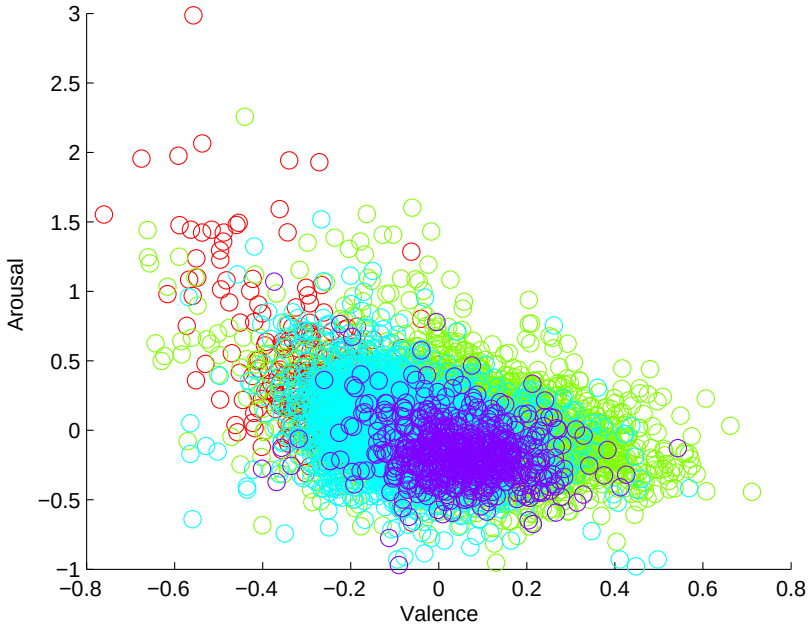


Figure 4.37: DTAKKC Linear Kernel

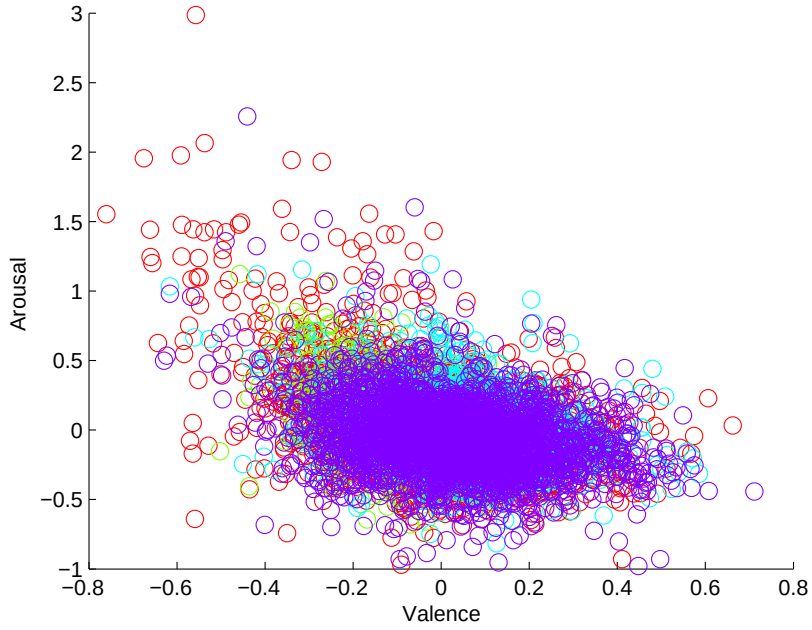


Figure 4.38: DTK

Figure 4.39: DTAKKC compared to DTK using 4 clusters performed on arousal and valence

Figure 4.40: Images extracted from different clusters: a) Bottom right: Purple cluster, Grandmother's Kitchen b) Bottom left: Blue Cluster, The Betrayal c) Top Right: Green Cluster, The Race b) Top Left: Red Cluster, Metro Goldwyn Mayer

the right exhibit everyday positive scenes with high valence. The difference in arousal is apparent in the left section: the top image is a high arousal scene of an individual running and the bottom scene is a low arousal scene of an individual cooking. The difference in arousal is not so apparent on the left images. This may be due to the asymmetrical distribution of the samples. Many of the samples have a disproportionate radial distance from the center onto the low valence and high arousal direction. This may be an interesting area to explore in future.

Table 4.11: Example films from different clusters with corresponding database indexes

Cluster	Films and database index
Purple (Happiness)	Becketts War (397-412), In the Mix (600-614)
Purple (Happiness)	The Home Coming (956-970), Grandmother's Kitchen (529-543)
Blue (Sadness, anger)	The Betrayal (413-442),Gustavo the Great (545-547))
Blue (Sadness, anger)	Then Doll And The Man Dog (911-9124), Chatter (1590-1615)
Green (Surprise,Fear)	Beautiful Sexy Funny Evil (384-396),The Race (1055-1037)
Green (Surprise,Fear)	The Robbery (1055-1071), Between Viewing (1301-1338)
Red(Fear, Agger)	The Room of Franz Kafka (8895-8907), Yembe (9638-9667))
Red(Fear, Agger)	Metro Goldwyn Mayer (1912-1968)

Changing the RBF parameter also leads to some interesting results. Fig. 4.41 shows three

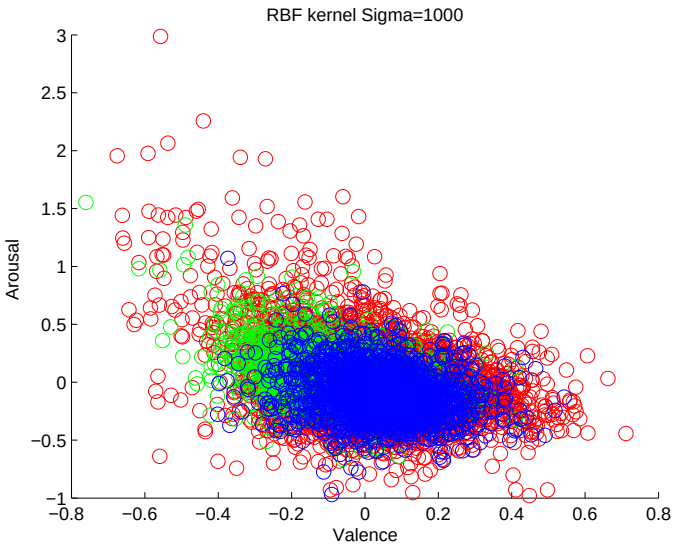


Figure 4.41: DTAKKC using RBF kernel with free parameter equal to 1000.

clusters using the DTAKKC method using both valence and arousal. The RBF kernel has a free parameter equal to 1000, and it seems the cluster membership is determined by the arousal axis. Fig. 4.42 shows the exact same procedure performed with the RBF kernel, with the only difference being that the free parameter is equal to 1. In this case it seems the cluster membership is determined by the valence axis, and this kind of flexibly gives the method a marked advantage over other methods .

4.4.3 Results: Linking Functions

This section demonstrates the ability of the segmentation linking function to segment areas in the VA space associated with scary movies. Fig. 4.43 shows three cluster DTAKKC linear kernels, and the sequences that have samples in the regions associated with violence are grouped together. Fig. 4.44 shows the results of the clustering using the region segmentation linking function, with the region indicated by a rectangle in the upper half of the VA space. It is evident that series that have any samples in a specified region are clustered together. These series can be flagged as containing content that is scary or clustered further. In addition, the method still maintains the

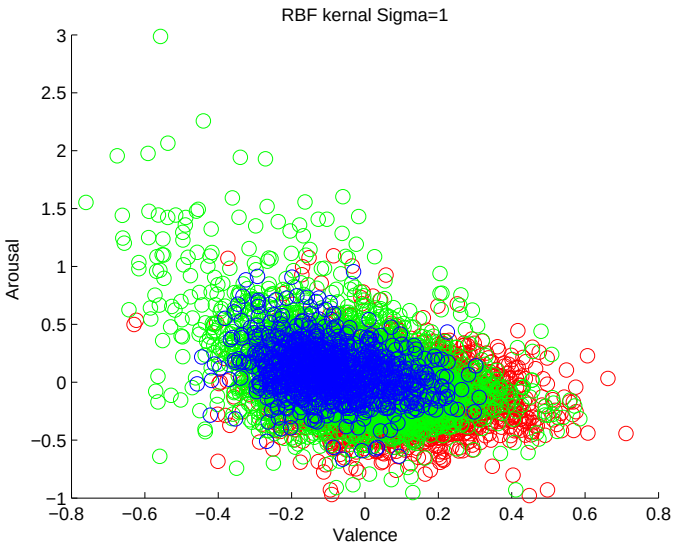


Figure 4.42: DTAKKC using RBF kernel with free parameter equal to 1.

valence ranking property.

4.5 Conclusion

This chapter develops a method to classify a two-dimensional valence arousal time series generated from a movie. A time series of features are extracted from a video sequence and mapped to the valence arousal plane. The method developed here performs a novel clustering method on a set of movies and clusters the entire movie sequence. The method is novel in that it uses time-alignment kernel operation with kernel k-means. The method is found to perform better than other state-of-the-art clustering methods. Additionally, the chapter tests different regression methods' abilities to map the low-level features of a video sequence onto the 2D emotion space using different types of regression.

Along with the linking function there are several other methods to adapt the kernels that can be used. The standard kernel operations can be applied to crate new Kernels, for example adding and multiply different Kernels. These operations can be incorporated directly into the Time warping operations or performed after the DTWK has been calculated. Segmenting the series before the

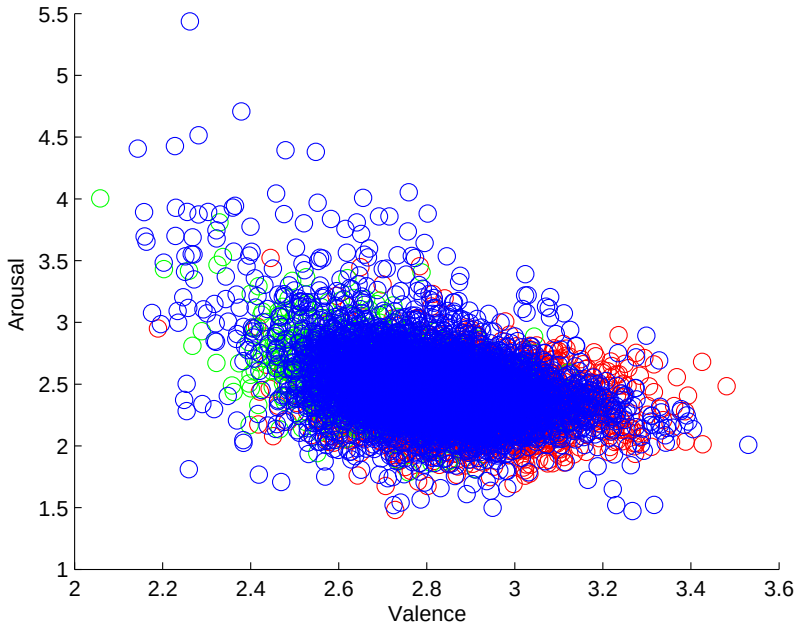


Figure 4.43: DTAKKC three cluster linear kernel.

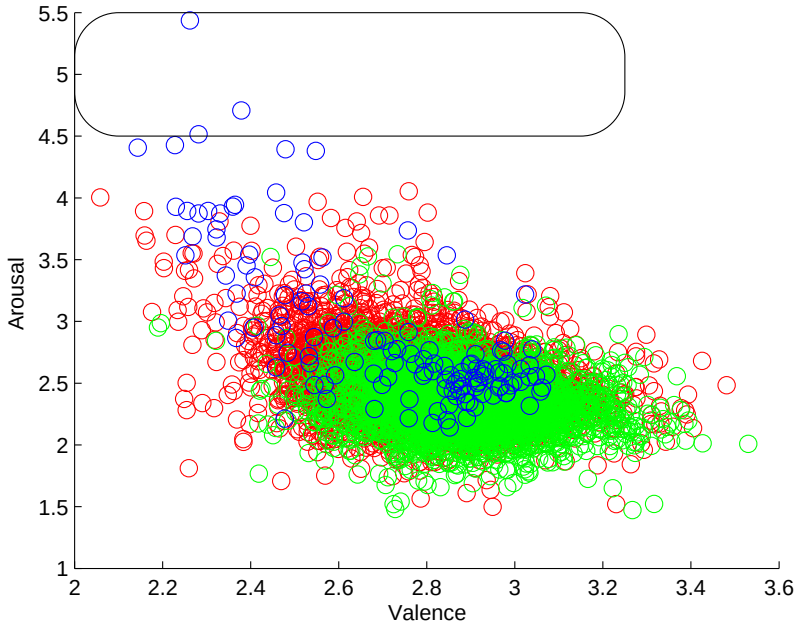


Figure 4.44: DTAKKC with link three using region segmentation linking function, region represented with box.

time warping operation can also be performed.

DTWK can be incorporated into kernel probabilistic clustering methods such as Kernel Trick Embedded Gaussian Mixture models [137], thus providing a likelihood. This likelihood could be used to calculate the posterior probability of the centroids [138] and the free kernel parameters and linking function values could be viewed as hyper-parameters. A prior distribution over the kernel parameters could also be determined and a posterior distribution could also be calculated. The prior distribution could be selected based on some pre-defined criteria. The main problem with these methods is that they are not tractable and require sampling methods or variational inference.

Chapter 5

State-Based Methods for Prediction of Valence and Arousal

In the previous chapters, we demonstrated that valence and arousal could be used to predict impact on cognition and that there was a correspondence between events and cognition. In this chapter, we develop several novel models to better predict valence and arousal that can be used to determine contents impact on cognition.

5.1 Introduction

In this section, we focus on two state-based models for mapping features on the valence and arousal plane. The main contribution is developing several state based models that better predict valence and arousal. These models either have overall better performance than the most popular models used in predicting valence and arousal in a specific genre or have overall better performance than models in the same class.

The idea is simple: emotions are dependent on what emotional state an individual is in, and knowing the emotional state will help better predict points on the valence and arousal plane. Most of the methods in [64, 66, 65, 125], focus on sparsity by constraining the solution using a prior assumption on the parameters. The main problem with these methods is that they use the same parameters for different emotional states. This is a problem because similar features can have a different mapping.

The first model works by associating the state with an event in a sports video. The main

advantage of this model is that these state can be determined in an unsupervised fashion and can also be used to detect sporting events that may influence the state. The main drawback is that the states' definitions are not as clear as there is not a true dependency. A true dependency assumption, such as input output HMM (IOHMM) [139], is not robust as it requires the use of the generalized expected maximization algorithm, and this is extremely difficult in the M-step for more than two states. The new method is called the dynamic prediction hidden Markov models (DPHMM)and uses HMM [55] architecture.

HMM has been used in semantic analysis for sports videos [140, 141, 48, 142]. We focus on arousal for sports videos because it is simpler to annotate. As LIRIS does not have a sports database we record the ATC used by self-assessment using the free software in [143]. This is achieved by manually annotating each frame of a video in real time using software [143]. The latter method is also tested with toy-data.

The second model uses mixtures of experts (ME), a type of neural network [89]. In this method each state represents different valence or arousal levels and is dependent on the feature vectors. The main advantage of ME over other methods is that it is not a black box and has some inherent meaning. In this case, the state z represents the response to some excitation or input $\mathbf{x}$ that changes the response.

One problem with ME is over-fitting. Unlike other neural networks, there is not a large body of work on regularization, and this leads a large number of features to produce incorrect results; a kernel representation would improve the method by giving it all the advantages associated with kernels. This can avoid dealing with large numbers of parameters, reduce the amount of computations in a high-dimensional space, and allow the use of infinite dimensionality feature spaces. As such, we formulate the ME model for linear regression method so that kernels can be used to avoid the explicit introduction of the feature vector. This model outperforms the standard ME model and has comparable performance to other state of the art regression models for prediction of valence and arousal. The supervised version for the code developed for this thesis is given in [144]. For the unsupervised method the gate values are obtained using [89].

5.2 Dynamic Prediction-Hidden Markov Models

In sports videos, one expects the arousal curve to follow a consistent pattern. Events elicit an arousal pattern similar to that of events in sports. We expect that a subject's arousal is dependent on what arousal state he or she is in. Therefore, we determine parameters for each state. This makes the estimated curve exhibit more comparability because each arousal event has its own set of coefficients. In addition, partitioning the data into states makes the curve smoother because there is more similarity between events.

The method functions by finding the most probable scalar dependent output based on a set of explanatory observations, using the Viterbi state sequence (VSS) and probabilistic modelling. The parameters are estimated using a novel formulation of the expected maximization algorithm [87]. The features used are those originally developed by [2], and include motion, rhythm, and sound energy. The method performs better in predicting the affective measure of arousal when compared to linear regression (LR), ridge regression (RR), Gaussian process regression (GPR) and relevance vector machine (RVM), which are used in the previous chapter with the same acronyms. Test measures include the residual squared error and visual assessment based on the original criteria given by [2]. The method is tested with simulated data and real data. Experimental results show that DPHMM outperforms the state-of-the-art in all criteria for most of the sports videos.

5.2.1 Problem Formulation: DPHMM

Assume there are K_{HMM} arousal states s_j $j = 1,..K_{HMM}$. Let $y_{t,j}$ represent the arousal of a subject produced by video frame $t \in \mathcal{Z}$. The arousal at state s_j can be modeled by:

$$y_{t,j} = \mathbf{w}_j{}^T \boldsymbol{\phi}(\mathbf{o}_t). \tag{5.1}$$

The vector $\mathbf{o}_t \in \Re^d$ is a feature vector containing the explanatory variables at frame t. As before, the basis functions are given by $\boldsymbol{\phi}(\mathbf{o}_t)$. The parameters $\mathbf{w}_j$ will map the observations to an arousal value and will be referred to as the set of state coefficients. Let q_t represent the hidden state variable, at some time t the endogenous variable Γ_t can be given by:

$$\Gamma_t = y_{t,j} + \xi_{t,j} | q_t = s_j. \tag{5.2}$$

The observation noise $\xi_{t,j} \sim \mathcal{N}(0, \sigma_j^2)$ is different for every state with noise variance σ_j^2. The underlying assumption is that the feature vector mapping is dependent on what arousal state the viewer is in. The method is different from the standard linear regression methods in that it assumes there are different arousal states for different arousal events. Using the assumption that the noise term is different for every state, one can formulate a probability density function for state s_j given by:

$$P(\Gamma_t|\mathbf{w}_j{}^T\boldsymbol{\phi}(\mathbf{o}_t), \sigma_j^2) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(\frac{-(\Gamma_t - \mathbf{w}_j{}^T\boldsymbol{\phi}(\mathbf{o}_t))^2}{2\sigma_j^2}\right) \tag{5.3}$$

The block diagram of the generation of Γ_t and the directed graph are shown in Fig. 5.1 A) and Fig. 5.1 B) respectively.

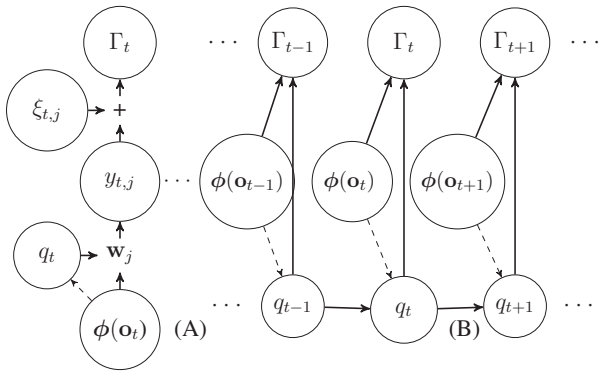


Figure 5.1: A) block diagram of the generation of Γ_t B) Graphical model of the probabilistic dependencies of the random variables, dashed lines are to indicate the state is dependent on observation via equation 5.4), but not a true dependency

The Model: DPHMM

It is well known that events in sports videos follow a sequence. Therefore, the arousal curve associated with these events should follow a similar sequential pattern. As a result, the model developed here is based on the entire observation sequence denoted by $\mathbf{O} : \{\mathbf{o}_1, .., \mathbf{o}_T\}$, where T is the length of the sequence. To determine the most likely time series $\mathbf{Y} : \{Y_1, .., Y_T\}$, the model depends on the most likely arousal sequence $Q^* = \{q_1^*, .., q_T^*\}$. Let $Y_t|\mathbf{O}$ be defined by:

$$Y_t|\mathbf{O} = \sum_{j=1}^{K_{HMM}} y_{t,j} I[q_t^* = s_j]. \tag{5.4}$$

The indicator function $I[q_t^* = s_j] = 0$ unless $q_t^* = s_j$ in that case $I[q_t^* = s_j] = 1$. The observation coefficients are different for every state s_j. The state path Q^* is the most probable hidden state sequence. To solve for Q^* an algorithm similar to the Viterbi algorithm [55] is used. The diagram of the process is shown in Fig. 5.2.

In order to determine the VSS, an emission distribution is needed. The emission distribution of Γ_t at state j is given by $\mathcal{N}(\Gamma_t|\mu_j, \epsilon_j^2)$ where μ_j ,ϵ_j^2 are the mean and variance respectively. As the variable Γ_t is not available, the estimated parameters for that state are used to determine the state likelihood :

$$P(\Gamma_t = y_{t,j}|\mu_j, \epsilon_j^2) = \frac{1}{\sqrt{2\pi\epsilon_j^2}} \exp\left(\frac{(y_{t,j} - \mu_j)^2}{2\epsilon_j^2}\right). \tag{5.5}$$

This provides the likelihood of the generated value $y_{t,j}$ at state j. The arousal state transitional probability coefficients $\hat{a}_{ij} = P(q_t = s_j|q_{t-1} = s_i)$ with an associated transitional probability matrix A and initial state distribution: $\hat{\pi}_j = P(q_1 = s_j)$, where $\boldsymbol{\pi} = [\hat{\pi}_1, .., \hat{\pi}_{K_{HMM}}]$ are also used to determine the likelihood of a state transition. The VSS can now be calculated by:

$$Q^* = \arg\max_Q \{\prod_t^T \frac{1}{\sqrt{2\pi\epsilon_{q_t}^2}} \exp\left(\frac{(y_{t,q_t} - \mu_{q_t})^2}{2\epsilon_{q_t}^2}\right) a_{q_{t-1},q_t}\pi_{q_1}\}. \tag{5.6}$$

Due to use of $y_{t,j}$ the state is a function of the input $\mathbf{o}_t$ therefore a dashed edge is added in Fig. 5.2. This is not a true probabilistic dependency, but an approximation. Methods do exist to generate a true probabilistic dependence like the Input Output HMM [145], but this algorithm did not function properly for this data.

An example of the process is demonstrated in the trellis diagram in Fig. 5.2. The three states s_j are displayed one per row, corresponding to the different values of q_t. The column represents the time index for q_t. The method determines every value of the ATC for all the states, then the Viterbi algorithm selects the realization of the ATC that maximizes the probability of the model.

$$\mathrm{P}(\Gamma, \mathbf{Q}|\boldsymbol{\phi}(\cdot), \mathbf{O}, \theta) = \prod_{t=1}^{T} \prod_{(i,j)}^{K_{HMM}} \left(\frac{1}{\sqrt{2\pi\sigma_j^2}} exp\left(\frac{-(\Gamma_t - \mathbf{w}_j{}^T \boldsymbol{\phi}(\mathbf{o}_t))^2}{2\sigma_j^2} \right) \right)^{I[q_t = s_j]} (a_{ij})^{I[q_t = s_j \wedge q_{t-1} = s_i]} (\pi_i)^{I[q_1 = s_i]} \quad (5.8)$$

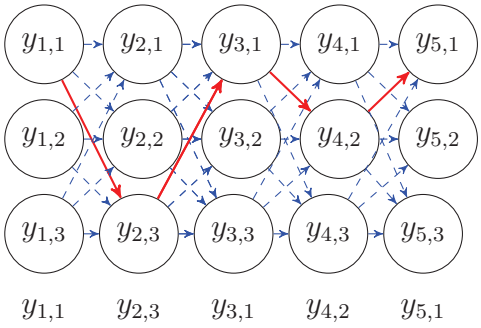


Figure 5.2: Trellis Diagram representing the relationship between the VSS in red and the time series for five observations and three states

5.2.2 Parameter Estimation: DPHMM

The conditional distribution of our model must also depend on the probability of a particular series of states Q. Thus, to train the model, we determine the set of parameters that maximize the likelihood of observing the mapping. From the observations $\mathbf{\Gamma} = [\Gamma_1, .., \Gamma_T]$ and the set of corresponding state labels $\mathbf{Q}$, the parameters θ in 5.7 can now be estimated by maximizing 5.8.

$$\theta = [A, \boldsymbol{\pi}, \mathbf{w}_1, .., \mathbf{w}_{K_{HMM}}, \sigma_1^2, .., \sigma_{K_{HMM}}^2] \quad (5.7)$$

In this case, the variables Q are not visible during training. Therefore, to estimate the parameters, the expectation maximization algorithm is used. Instead of maximizing the likelihood directly, we find the posterior distribution of the latent variables $P(Q|\mathbf{\Gamma}, \mathbf{O}, \boldsymbol{\phi}(\cdot), \theta_{l-1})$. This posterior distribution is used to evaluate the expectation of the logarithm of the complete data likelihood function or equivalently maximizing:

$$\tilde{l}(\theta, \theta_{l-1}) = \sum_{Q} P(Q|\mathbf{\Gamma}, \mathbf{O}, \boldsymbol{\phi}(\cdot), \theta_{l-1}) \log(P(\mathbf{\Gamma}, \mathbf{Q}|\boldsymbol{\phi}(\cdot), \mathbf{O}, \theta)) \quad (5.9)$$

The term θ_{l-1} is the previous estimate at index $l-1$. The parameters are randomly initialized.

Once the standard stochastic constraints are added to 5.8, the parameters can be determined with the following equations:

$$\hat{\pi}_i = P(q_1 = s_i|\boldsymbol{\phi}(\cdot), \mathbf{O}, \theta_{l-1}). \tag{5.10}$$

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T} P(q_{t-1} = s_i, q_t = s_j|\boldsymbol{\phi}(\cdot), \mathbf{O}, \theta_{l-1})}{\sum_{t=1}^{T} P(q_{t-1} = s_i|\boldsymbol{\phi}(\cdot), \mathbf{O}, \theta_{l-1})} \tag{5.11}$$

$$\Phi_j = \sum_{t=1}^{T} \boldsymbol{\phi}(\mathbf{o}_t)\boldsymbol{\phi}(\mathbf{o}_t)^T P(q_t = s_j|\boldsymbol{\phi}(\cdot), \mathbf{O}, \theta_{l-1}) \tag{5.12}$$

$$\hat{\mathbf{w}}_j = \Phi_j^{-1} \sum_{t=1}^{T} \Gamma_t \boldsymbol{\phi}(\mathbf{o}_t) P(q_t = s_j|\boldsymbol{\phi}(\cdot), \mathbf{O}, \theta_{l-1}) \tag{5.13}$$

$$\hat{\sigma}_j^2 = \frac{\sum_{t=1}^{T} (\Gamma_{t,j} - \omega_{\mathbf{j}}{}^T \boldsymbol{\phi}(\mathbf{o}_t))^2 P(q_t = s_j|\boldsymbol{\phi}(\cdot), \mathbf{O}, \theta_{l-1})}{\sum_{t=1}^{T} P(q_t = s_j|\boldsymbol{\phi}(\cdot), \mathbf{O}, \theta_{l-1})}. \tag{5.14}$$

In addition, the mean and variance of Γ_t at each state s_j is determined for the emission distributions:

$$\hat{\mu}_j = \frac{\sum_{t=1}^{T} \Gamma_t P(q_t = s_j|\boldsymbol{\phi}(\cdot), \mathbf{O}, \theta_{l-1})}{\sum_{t=1}^{T} P(q_t = s_j|\boldsymbol{\phi}(\cdot), \mathbf{O}, \theta_{l-1})} \tag{5.15}$$

$$\hat{\varepsilon}_j^2 = \frac{\sum_{t=1}^{T} (\Gamma_t - \hat{\mu}_j)^2 P(q_t = s_j|\boldsymbol{\phi}(\cdot), \mathbf{O}, \theta_{l-1})}{\sum_{t=1}^{T} P(q_t = s_j|\boldsymbol{\phi}(\cdot), \mathbf{O}, \theta_{l-1})}, \tag{5.16}$$

where Φ_j is an intermediate term for calculation. The EM algorithm only finds local maximums, therefore multiple initializations are performed and the initialization with the largest likelihood is used. When calculating the probabilities, the backward and forward algorithms and scaling factors are used [55]. A Matlab implementation for demonstration of the training process with a regularization modification is available at [146].

As in the previous section, we used RSE to evaluate the algorithm. For this section the RSE is defined in 5.17. In addition to the RSE was used and the criteria given by [2].

$$RSE = \sum_{t=1}^{T} (\Gamma_t - Y(t))^2 / \sum_{t=1}^{T} \Gamma_t^2 \tag{5.17}$$

Feature Space

In this section we describe the arousal features. We select these features because they have been shown to have a correlation with arousal [66], and were developed by [2]. The term $\boldsymbol{\phi}(\mathbf{o}_t) = \mathbf{o}_t$ with components $(\mathbf{o}_t)_d = o_{d,t}$.

Each video frame's motion vectors $\mathbf{v}_{k,t}$ are computed using the standard M_b block-based motion estimation between two adjacent frames t and $t-1$. The motion activity is given by:

$$o_{1,t} = \frac{1}{\underset{\ell}{max}\{|\mathbf{v}_{\ell,t}|\}} \sum_{m_b=1}^{M_b} |\mathbf{v}_{m_b,t}|. \tag{5.18}$$

The rhythm component is a time-varying function proportional to the time difference between preceding and succeeding shots. Let $\alpha(t)$ be the frame index of the preceding shot of frame t, and let $\beta(t)$ be the index of the succeeding shot of frame t . The rhythm of shot t is given by:

$$o_{2,t} = e^{1-(\beta(t)-\alpha(t))}. \tag{5.19}$$

Sound energy is a classic affective feature. The sampling rate of audio is so much larger than video, and therefore one video frame will have multiple samples of audio. These samples are also referred to as frames. Let $s_t[n]$ be the audio sample of the $t-th$ frame of video with N_s samples. The sound energy is given by:

$$o_{3,t} = \sum_{n=1}^{N_s} (s_t[n])^2. \tag{5.20}$$

5.3 Block Diagram: DPHMM

The final block diagram is shown in Fig. 5.3 after feature extraction represented by the purple block a prediction for each state is made represented by the red block underneath. Then the Viterbi

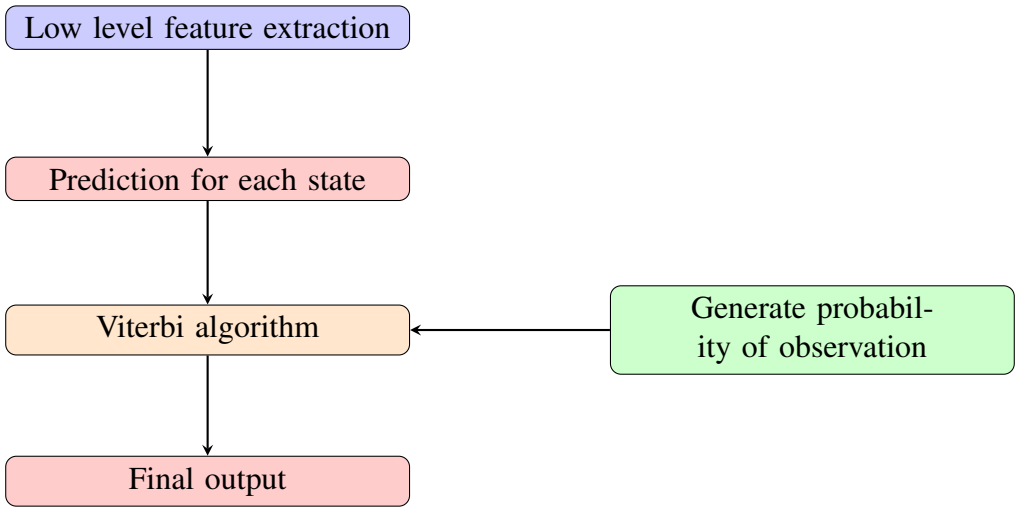
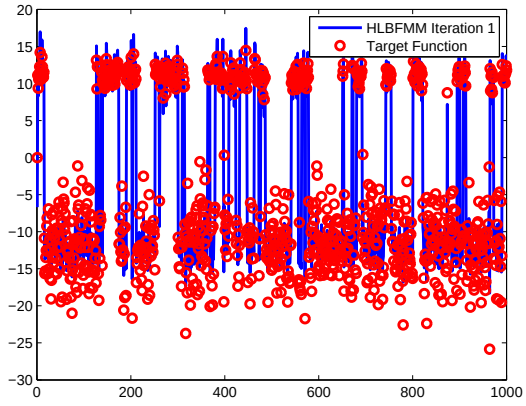


Figure 5.3: After feature extraction a prediction for each state is made. Then the Viterbi state sequence is calculated and the most probable states are used to generate an output.

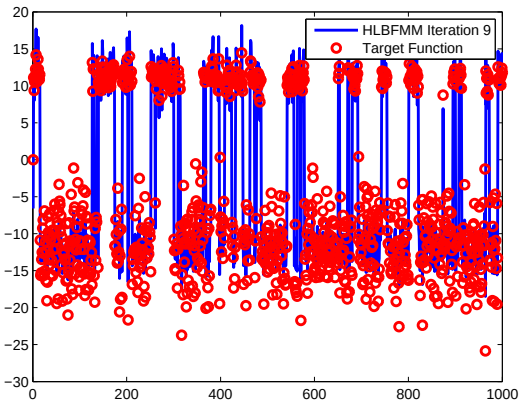
state sequence is calculated from the probability of observations and the most probable states are used to generate an output.

5.4 Toy data: DPHMM

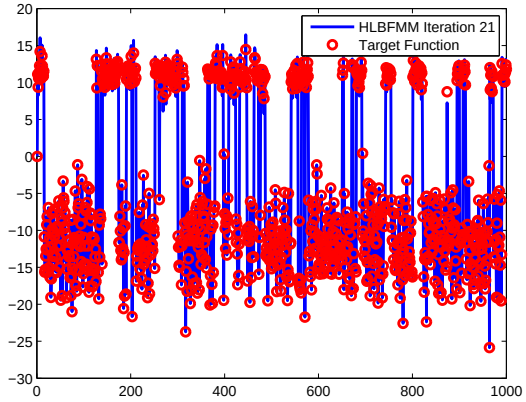
To better understand the convergence properties of the algorithm, we investigate the relationship between the RSE and the log-likelihood with simulated training data. In Fig. 5.4, we see different realizations of $Y(t)$ in blue for different iterations of the EM algorithm, with the target values overlaid in red. The code for the toy data is available at [146]. It is evident for every iteration that the data is better fitted, with the most marked improvement exhibited between the $9th$ and $20th$ iteration. In Fig. 5.5, we see the corresponding learning curve for the data; it is evident that as the log-likelihood increases the RSE decreases. The largest change occurs between the $10th$ and $20th$ iteration, which corresponds to the results in Fig. 5.4.



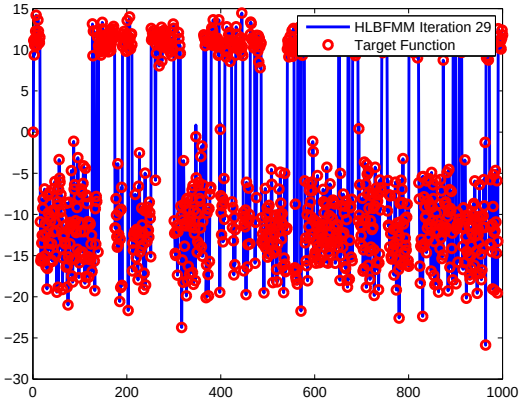
(a) first iteration



(b) $9th$ iteration



(c)$20th$ iteration



(d)$29th$ iteration

Figure 5.4: Example of Y overlaid of target values of training data after different iteration of EM algorithm.

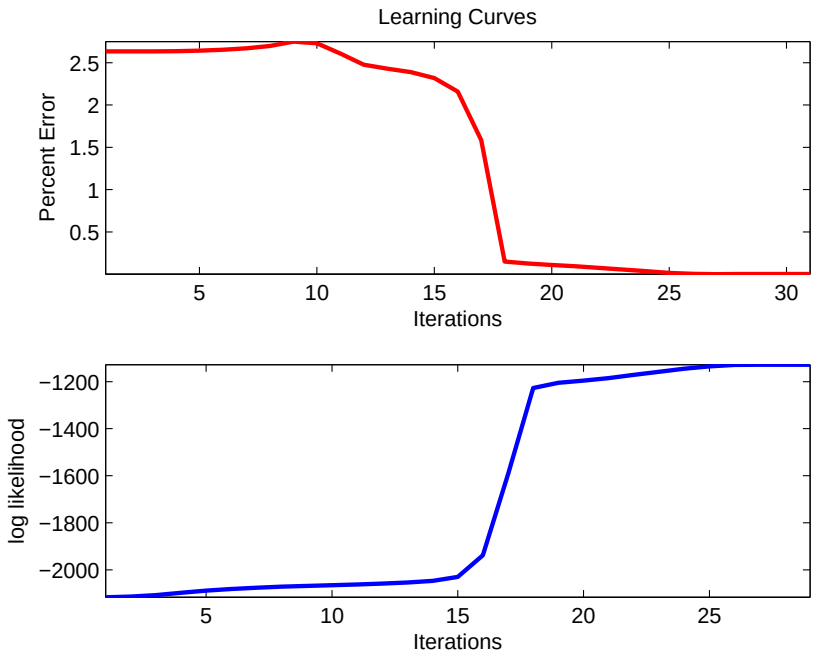


Figure 5.5: Learning curves for different iteration of EM Algorithm *Top*: RSE for different iterations of EM Algorithm. *Bottom*: log-likelihood of model using the estimated parameter values from each iteration of the EM Algorithm .

5.4.1 Experimental Procedure: DPHMM

To train the system, six participants are asked to annotate the ATC of one of the sports videos for thirty minutes. In affective systems only one participant is required [64]. The experimental setup is shown in Fig. 5.6. The screen on the top left displays the video. The scroll bar on the right is controlled by the mouse and annotates the arousal. The annotation software records the position of the scroll bar at a sampling rate much larger than that of the frame rate of video. The annotation signal is then down-sampled to that of the frame rate of the video and manually aligned due to delays in the annotation software of the user. Some regions have been discarded as it is evident that the user lost interest or stopped controlling the mouse correctly.

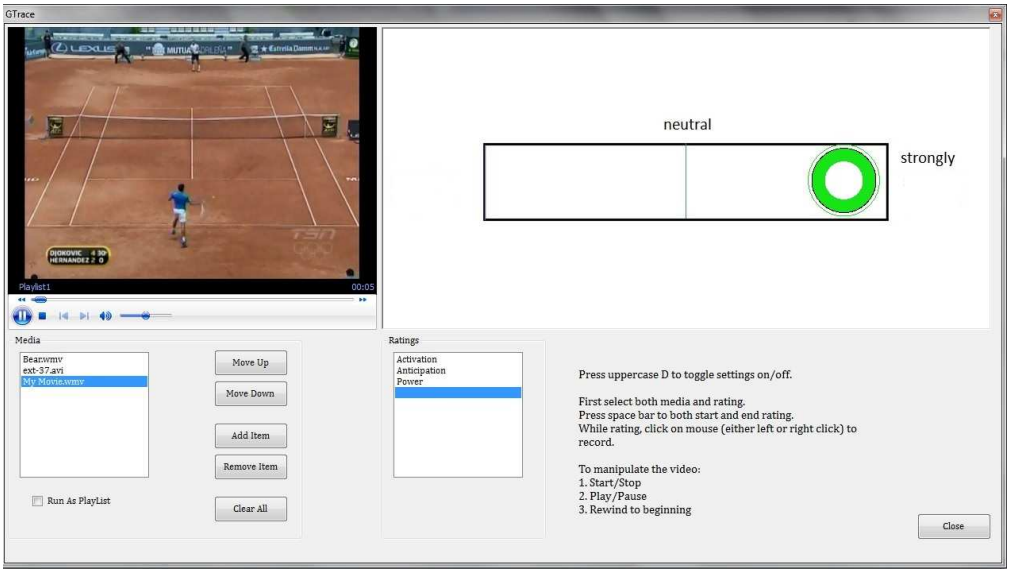


Figure 5.6: Experimental setup: The screen on the top left is the displayed video and the scroll bar on the right is used to record the arousal levels.

The data-set is taken from several sources: 2009 PGA Tour, 2009 World Series of Bowling, 2009 PDC World Darts Championship, 2009 Wimbledon Championship, 2006 FIFA soccer World Cup, and NHL hockey game 2010. The signal was then aligned and down sampled. The individual is asked to annotate a ten minute clip and then two-fold cross validation is used to determine the average testing error, using half the signal for training and half for validation.

5.4.2 Results: DPHMM

An example of the annotation variation among two different participants is shown in Fig. 5.7. It is evident that the arousal exhibits similar characteristics and there is a correspondence with rhythm. The similarity is probably due to the fact that exciting events occur in certain intervals, but there are some duration and amplitude differences.

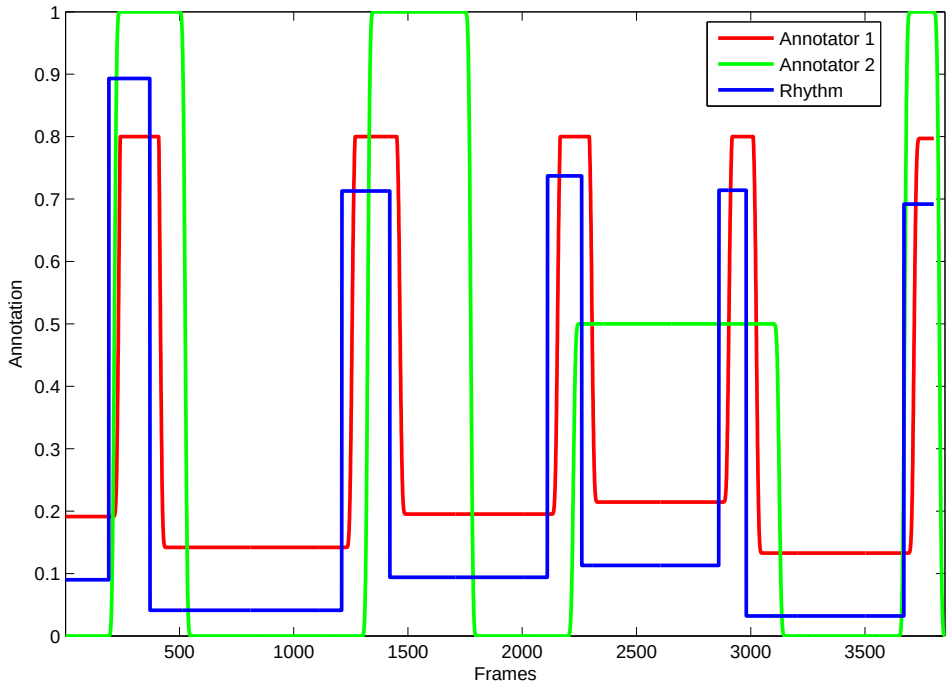


Figure 5.7: Annotation among different participants for the same video with Rhythm shown for demonstration.

Table. 5.1 shows the RSE for the different methods. It is evident, with the exception of tennis, that the new method outperforms state-of-the-art methods, such as RVM. The IOHMM was also included using the Softmax as a gate, if the result was unstable it will be denoted by a $\times$. It is also evident from Table. 5.1 that the novel method also substantially outperforms the other methods. In the tennis videos, all methods perform badly, with all methods having errors over 475%. This is most likely due to the fact that tennis has less audio and motion. Four states were optimal for most sports, while in bowling two states performed best.

Fig. 5.8 A) demonstrates how the DPHMM outperforms RVM in the psychological criteria. The actual arousal curve is in red, with the peaks in the curve representing when the bowler takes a shot. The blue curve is estimated using DPHMM and the green curve is estimated using RVM. Fig. 5.8 B) displays the state for each frame; in most cases the state changes correspond to the peaks. RVM is used for comparison because it has the second best performance. It is self-evident that DPHMM is considerably better in meeting the smoothness criteria. Comparing compatibility,

Table 5.1: RSE for different methods and sports videos on test samples

Method	LR	RR	GPR	RVM	IOHMM	DPHMM
Golf	0.1653	0.1653	2.54	0.169	1.64	**0.130**
Bowling	3.85	3.85	4.48	0.792	×	**0.743**
Darts	0.961	0.958	1.36	0.209	×	**0.162**
Tennis	49.3	49.2	8.35	**4.76**	×	6.36
Hockey	0.253	0.252	4.35	0.22	0.53	**0.193**
Soccer	0.5812	0.5812	8.35	0.5712	×	**0.5387**

the DPHMM has a more parabolic-like shape. Both curves generate negative values outside the specified range, but these values are less frequent and closer to the specified range using DPHMM than RVM. There is a strong relationship between the rhythm shown in Fig. 5.7 and the generated curve in Fig. 5.8. This is likely the case because the rhythm component encodes duration of events that have a predictable arousal response.

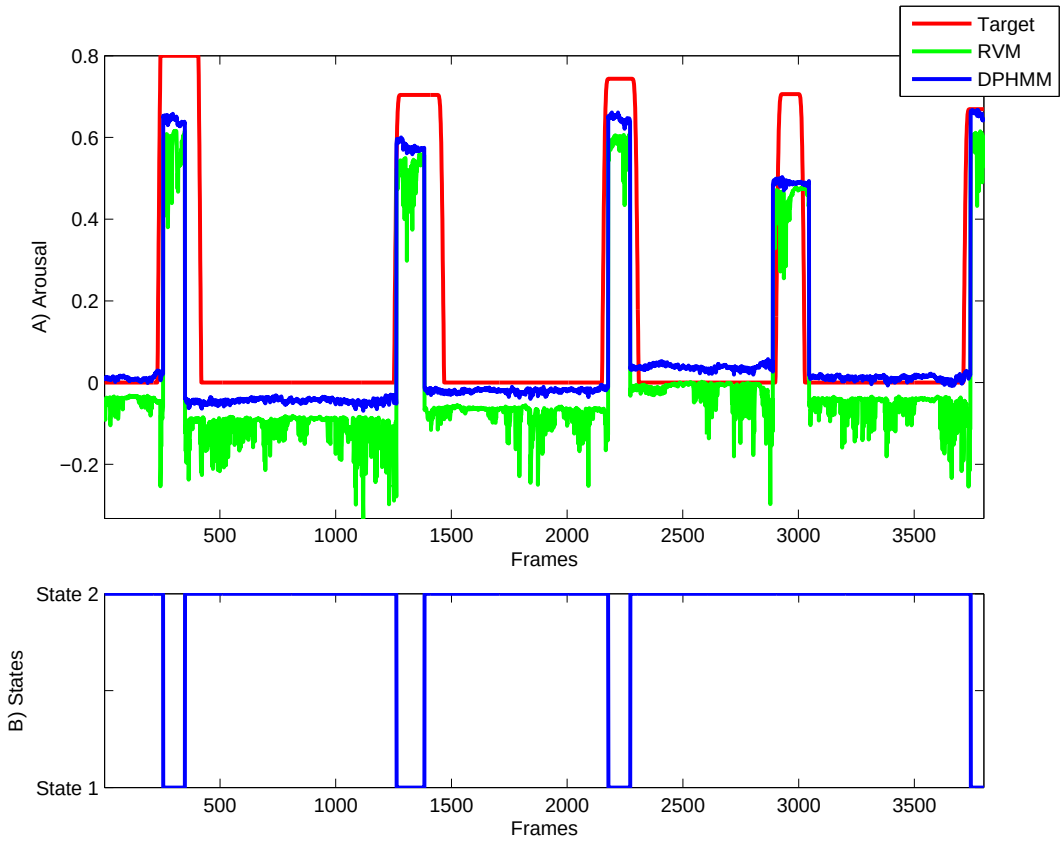


Figure 5.8: A)Target data overlaid by smooth curve estimated by DPHMM and jagged curve developed by RVM B) States

To compare compatibility, we see the values for the curve generated by DPHMM are much more uniform and in most cases much closer to the actual arousal curve, with the exception of the video frames around the 3000 mark. Examining Fig. 5.8 B), we see that no state transition occurs, thus the wrong state coefficients were used in the mapping. The model seems to work in data that has sudden jumps. Another issue is that the model seems to suffer from numerical issues as the dimension of the data gets larger, even if toy data is used. One fact to take into consideration is the random initialization of the EM algorithm, which may be aggregating the data and somehow acting as a kind of bootstrap sampling. This has been shown to reduce error [147].

5.5 Kernel-Based Mixture of Experts

Mixture of experts output is dependent on the state as well as the input and uses a precise probabilistic definition. Unlike the DPHMM, the kernel-based mixture of experts has no dependence assumption between states, but the dependency between observation and state is direct as shown in Fig. 5.9. The variable z is a latent variable that models some hidden process that produces valence or arousal given the feature $\mathbf{x}$. The process models the valence or arousal Γ by assuming that the state of the individual z and the input features $\mathbf{x}$ both play a role in the estimation.

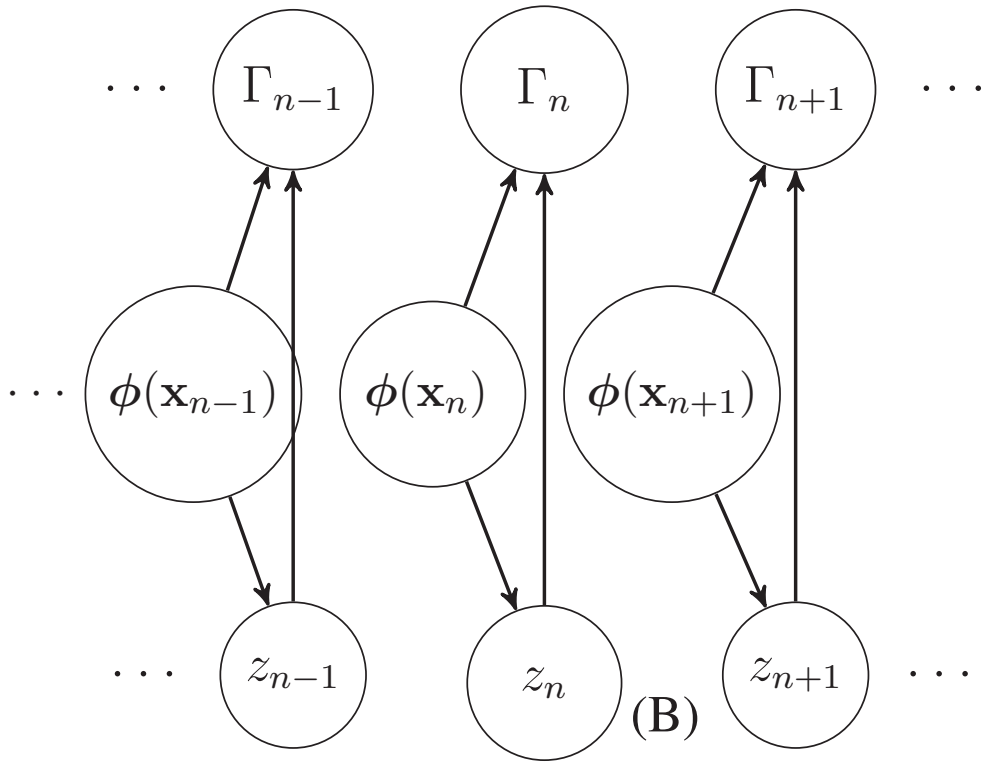


Figure 5.9: Graphical model of the probabilistic dependencies of the random variables for ME

Similar to the previous section, we verify our model using simulated data, then compare the model to the classical ME for linear regression. It is observed that the novel method has increased performance for predicting valence and arousal when compared to regular ME. Most notably, the

RBF kernel performs well. The method is tested by using toy data, and we find that it provides better numerical stability than the classic method for polynomial data as the amount of data decreases relative to the order of the polynomial. The results show that the novel method performs better than the classical ME for linear regression in predicting arousal and valence. Kernels tested include linear, polynomial and RBF kernels.

The main contribution of this section is formulating the ME for regression using kernels, which has several advantages. Perhaps the most obvious is the ability to take advantage of all the research related to kernels and a new framework for applying kernels. There are several advantages particular to mixture of experts. The first is that kernels provide a way to combat the curse of dimensionality, a problem which is exacerbated with mixture of experts because the problem is multiplied by the number of experts. Another advantage is computational savings. Similar to the aforementioned problem, any computational cost is multiplied by the number of experts. Finally, there is a closed form solution for a maximum, which is extremely important in the unsupervised case because optimization methods are much more complex for unlabeled data. In this chapter, we will use a supervised version for simulated data and an unsupervised version for the arousal and valence estimation. To simplify the optimization we will use the method in [89] for the gates. An open source implementation developed for this thesis of [89] is available at [144].

5.5.1 Problem Formulation: Kernel-Based Mixture of Experts

Let $y_k(\mathbf{x})$ be the output of the $k-th$ expert, given some input $\mathbf{x}$. In order to make a single prediction, the output of the ME architecture is given by:

$$y(\mathbf{x}) = \sum_{k=1}^{K_m} p(z = k|\mathbf{x}, \mathbf{v}_k) y_k(\mathbf{x}), \tag{5.21}$$

where z is a variable indicating the expert used, and $p(z = k|\mathbf{x}, \mathbf{v}_k)$ is the gating probability density function. This provides an indication of the likelihood of expert $y_k(\mathbf{x})$ contributing to the output. The gaining function segments the input space accordingly and has the parameters $\mathbf{v}_k$. In the classical ME for regression, the function $y_k(\mathbf{x})$ like DPHMM is:

$$y_k(\mathbf{x}) = \mathbf{w}_k^T \boldsymbol{\phi}(\mathbf{x}). \tag{5.22}$$

The function $y_k(\mathbf{x})$ is explicitly a function of $\mathbf{x}$; the variable $\mathbf{w}_k$ is simply a parameter. In this section we will show that 5.22 can be expressed as a kernel function, symbolically:

$$y_k(\mathbf{x}) = \mathbf{k}(\mathbf{x})^T(\mathbf{\Lambda}_k\mathbf{K} + \lambda_k I_N)^{-1}\mathbf{\Lambda}_k\mathbf{\Gamma}. \tag{5.23}$$

Where the vector $\mathbf{\Gamma} = [\Gamma_1, ., \Gamma_N]$ is the set of targets. As before $\mathbf{K} \in \Re^{NxN}$ the Gram matrix with elements $(\mathbf{K})_{i,j} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ and $\mathbf{k}(\mathbf{x})$ is a vector with elements $(\mathbf{k}(\mathbf{x}))_j = \kappa(\mathbf{x}, \mathbf{x}_j)$. The matrix I_N is a NxN regularization matrix and we define λ_k as the regularization term for the $k-th$ expert. Finally we define the indicator matrix $\mathbf{\Lambda}_k$ for the $k-th$ expert as:

$$\mathbf{\Lambda}_k \equiv dig([I[z_1 = k], .., I[z_N = k]])$$

The indicator function $I[z_n \neq k] = 0$ unless $z_n = k$ in that case $I[z_m = k] = 1$. For this work the gating function will be the Softmax Function [148]:

$$P(z = k|\mathbf{x}', \mathbf{v}_k) = \frac{exp(\mathbf{v}_k^T\mathbf{x}')}{\sum_{k'=1}^{K_m} exp(\mathbf{v}_{k'}^T\mathbf{x}')}. \tag{5.24}$$

Where $\mathbf{x}'$ indicates the inclusion of a term for the bias, symbolically $\mathbf{x}' = [1||\mathbf{x}]^T$.

5.6 Estimation: Kernel-Based Mixture of Experts

5.6.1 Cost Function: Kernel-Based Mixture of Experts

Let Γ be the target function modeled by a deterministic function $y_k(\mathbf{x})$. Using maximum likelihood estimation (MLE) we can estimate the parameters. Taking the negative logarithm of the MLE, one can obtain the following cost function:

$$\hat{l}(\mathbf{W}, \mathbf{V}) = \sum_{k=1}^{K_{em}} -ln(\prod_{n=1}^{N} P(\Gamma_n, z_n|\boldsymbol{\phi}(\cdot), \mathbf{x}_n, \mathbf{w}_k, \mathbf{v}_k)^{I(z_n=k)}). \tag{5.25}$$

Where $\mathbf{W} = [\mathbf{w}_1, .., \mathbf{w}_{K_m}]$ are the parameters for the experts and $\mathbf{V} = [\mathbf{v}_1, .., \mathbf{v}_{K_m}]$ represent the parameters of the gating function. Decomposing equation 5.25 and rewriting the product in terms of summation one can obtain:

$$\hat{l}(\mathbf{W}, \mathbf{V}) = \sum_{(k,n)}^{(K_m,N)} -I(z_n = k)(ln(P(\Gamma_n|z_n, \boldsymbol{\phi}(\mathbf{x}_n), \mathbf{w}_k)) - ln(P(z_n|\mathbf{x}_n, \mathbf{v}_k))). \quad (5.26)$$

The second term depends on the gating function and can be optimized separately [149]. Using the standard normal assumption and considering only terms that depended on $\mathbf{w}_k$, the first term in equation 5.26 can be written as:

$$\hat{l}(\mathbf{W}) = \sum_{n=1}^{N}\sum_{k=1}^{K_m} I(z_n = k)\left(\frac{(\Gamma_n - \mathbf{w}_k^T\boldsymbol{\phi}(\mathbf{x_n}))^2}{2\sigma_k^2} + \lambda_k\frac{C_k}{2}\mathbf{w}_k^T\mathbf{w}_k\right). \quad (5.27)$$

With quadratic regularization term $\mathbf{w}_k^T\mathbf{w}_k$ added for numerical stability and as an artifice to introduce the kernel. The regularization constant λ_k is usually determined empirically. Therefore to simplify the expression in its final form we define C_k as:

$$C_k = \frac{1}{\sum_{n=1}^{N} I(z_n = k)\sigma_k^2} \quad (5.28)$$

Converting the expression in 5.27 to matrix form, one can obtain:

$$l(\mathbf{W}) = \sum_{k=1}^{K_m}(\boldsymbol{\Gamma} - \Phi\mathbf{w}_k)^T\frac{\boldsymbol{\Lambda}_k}{2\sigma_k^2}(\boldsymbol{\Gamma} - \Phi\mathbf{w}_k) + \frac{\lambda_l}{2\sigma_k^2}\mathbf{w}_k^T\mathbf{w}_k. \quad (5.29)$$

Where $\Phi^T = [\boldsymbol{\phi}(\mathbf{x}_1), \boldsymbol{\phi}(\mathbf{x}_2), ..., \boldsymbol{\phi}(\mathbf{x}_{N-1}), \boldsymbol{\phi}(\mathbf{x}_N)]$.

5.6.2 Classical Solution: Mixture of Experts

Before presenting the novel algorithm, the original formulation or the parametric solution is stated here for clarity. By taking the gradient of equation 5.29 with respect to $\mathbf{w}_k$, then setting the equations equal to zero and solving for $\mathbf{w}_k$, the following expression can be obtained:

$$\hat{\mathbf{w}}_k = (\Phi^T\boldsymbol{\Lambda}_k\Phi + \lambda_k I_N)^{-1}\Phi\boldsymbol{\Lambda}_k\boldsymbol{\Gamma}. \quad (5.30)$$

This is analogous to the primal solution in [94]. The matrix $(\Phi^T\boldsymbol{\Lambda}_k\Phi + \lambda_k I_N)$.

$$l(\mathbf{A}) = \sum_{k=1}^{K_{em}} \frac{1}{2\sigma_k^2}(\mathbf{\Gamma} - \Phi\Phi^T\mathbf{a}_k)^T\mathbf{\Lambda}_k(\mathbf{\Gamma} - \Phi\Phi^T\mathbf{a}_k) + C_k\frac{\lambda_k}{2}\sum_{n=1}^{N} I(z_n = k)\mathbf{a}_k^T\Phi\Phi^T\mathbf{a}_k. \quad (5.33)$$

5.6.3 Novel Solution

In this section, we formulate the novel solution so that kernels can be used, analogous to the dual solution in [94]. In the same manner as above, we can solve for $\mathbf{w}_k$. One can solve the problem using constrained optimization. For each of the expert parameters we can formulate a constrained optimization problem and minimize the unconstrained one using KKT. The unconstrained cost function is:

$$\Lambda(\boldsymbol{\xi}_1, .., \boldsymbol{\xi}_{K_{em}}, \mathbf{W}, \mathbf{A}) = \left\{ \sum_{k=1}^{K_{em}} \boldsymbol{\xi_k}^T\boldsymbol{\xi}_k + \alpha_p\mathbf{w}_k^T\mathbf{w}_k + \mathbf{a_k}^T(\mathbf{\Lambda}_k\mathbf{\Gamma} - \mathbf{\Lambda}_k\Phi\mathbf{w_k} - \boldsymbol{\xi_k}) \right\}. \quad (5.31)$$

As the gradients with respect to $\mathbf{w}_k$ in 5.31 are independent of each other, we can solve for each term in the summation independently using the same process as in 2.48- 2.58. It can be shown that:

$$\mathbf{w}_k = \frac{-1}{\lambda_k}\Phi^T\mathbf{\Lambda}_l(\Gamma - \Phi\mathbf{w}_k) = \Phi^T\mathbf{a}_k. \quad (5.32)$$

Substituting equation 5.32 back into equation 5.29 gives equation 5.33: Where $\mathbf{A} = [\mathbf{a}_1, .., \mathbf{a}_{K_{em}}]$. In the same manner as above it can be shown that the value of $\mathbf{a}_k$ that minimizes 5.33 is given by:

$$\mathbf{a}_k = (\mathbf{\Lambda}_k\mathbf{K} + \lambda_k I_N)^{-1}\mathbf{\Lambda}_k\mathbf{\Gamma}, \quad (5.34)$$

which is achieved by using the fact that $\mathbf{K} = \Phi\Phi^T$. We can now express $y_k(\mathbf{x})$ in terms of kernels, first by inserting equation 5.32 into equation 5.22:

$$y_k(\mathbf{x}) = \mathbf{w}_k^T\boldsymbol{\phi}(\mathbf{x}) = (\Phi^T\mathbf{a}_k)^T\boldsymbol{\phi}(\mathbf{x}) = \mathbf{a}_k^T\Phi\boldsymbol{\phi}(\mathbf{x}) = \mathbf{a}_k^T\mathbf{k}(\mathbf{x}) \quad (5.35)$$

Now inserting equation 5.34 into equation 5.35 and using the fact that $y_k(\mathbf{x})^T = y_k(\mathbf{x})$ the final form can be achieved:

$$y_k(\mathbf{x}) = \mathbf{k}(\mathbf{x})^T\mathbf{a}_k = \mathbf{k}(\mathbf{x})^T(\mathbf{\Lambda}_k\mathbf{K} + \lambda_k I_N)^{-1}\mathbf{\Lambda}_k\mathbf{\Gamma} \quad (5.36)$$

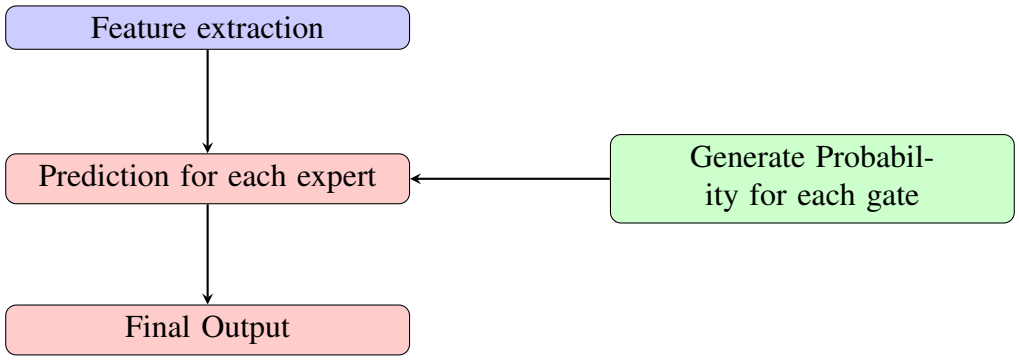


Figure 5.10: After feature extraction a prediction for each expert is made. This is combined with the output from each gate to produce a prediction.

5.6.4 Error

In order to investigate the properties of the algorithm, the residual squared error will be used on the test set. For convenience, we restate it here:

$$RSE = \sum_{n=1}^{N} (\Gamma_n - y(\mathbf{x}_n))^2 / \sum_{n=1}^{T} \Gamma_n^2. \tag{5.37}$$

5.7 Block Diagram: Kernel-Based Mixture of Experts

The final block diagram is shown in Fig. 5.10. After feature extraction represented by the purple block, a prediction for each expert is made represented by the red block. This is combined with the output from each gate shown with the green block to produce a final prediction.

5.8 Data Sets

5.8.1 Simulated Data: Polynomial Kernels

In order to validate the model and test robustness to numerical stability, we generate data in the feature space. We use the standard method of generating random toy data for regression in algorithm 1.Where $\mathbf{x}_n$ is uniformly distributed and linearly separable with a corresponding expert label $Z[m]$. We will evaluate the toy data for polynomial basis functions.

Algorithm 1 Algorithm: Generate Toy Data

Input:
$\{\mathbf{w}_1, .., \mathbf{w}_{K_m}\}$,$\{\sigma_1^2, .., \sigma_{K_m}^2\}$
Z: array of labels for data X
X:$\{\mathbf{x}_1, .., \mathbf{x}_n\}$
while $n \leq N$ **do**
  $n = n + 1$
  $k = 1$
  while $k \leq K_{em}$ **do**
    $l = l + 1$
    if $Z[n] = l$ **then**
      $\xi_n \sim \mathcal{N}(0, \sigma_k^2)$
      $\Gamma_n = \mathbf{w}_k^T \boldsymbol{\phi}(\mathbf{x}_n) + \xi_n$
    end if
  end while
end while

5.8.2 Real Data

We use the LIRIS as in Chapter 5. The same procedure is used, however in order to make computation feasible, we use less data in the training data when working with dual variables. This does not affect performance.

5.9 Experiments Results: Kernel-Based Mixture of Experts

5.9.1 Simulated Data

The method is tested for orders up to 3 and $K_m = 3$ using the parametric method and the novel method; two experts are used. Cross-validation is employed, using half the generated data for training and half for testing. The procedure is performed a total of 100 times and the average is then taken. Table. 5.2 shows the error of the toy data, where MER represents the classic ME for regression. Each row of the table represents the order of the polynomial function used and each column represents the amount of data used in training. The first entry in the table shows the average error using the kernel method, while the second entry shows the error using the standard ME model. It is evident when there are 50 samples of data that the two models are equivalent

with a maximum difference of error of 0.02%. As less data is used to train, both models degrade in performance, but this degradation is much smaller in the novel method. This degradation is more pronounced as d becomes larger, until eventually a singular matrix (SM) occurs in the design matrix in equation 5.30.

This does not occur in the novel method. The reduced rate of error in the novel method is because the Gram matrix dimensions decrease with the reduction in data. This makes the Gram matrix less susceptible to numerical instability. Additionally, the Gram matrix size is constant with respect to the order of the polynomial; the dimensions of the design matrix increases exponentially with the order of the polynomial. These observations are surprising because the toy data is generated using the basis function. The different RSE for the order two polynomials is shown in Fig. 5.11. The error seems to dramatically increase as the number of data points used in training decrease to 25, but this increase is much smaller with the kernel method.

Table 5.2: Average RSE for two methods where (novel method, MER) row represents order of the polynomial and column represents the amount of data

Samples	50	25	12	7
d=3	(0.25%,0.26%)	(0.29%,0.56%)	(25%,30%)	(20%,SM)
d=2	(0.90,0.88%)	(0.60%,0.53%)	(8%,13%)	(20%,SM)
d=1	(0.18%,0.17%)	(0.19%,0.211%)	(2.7%,3.9%)	(20%,SM)

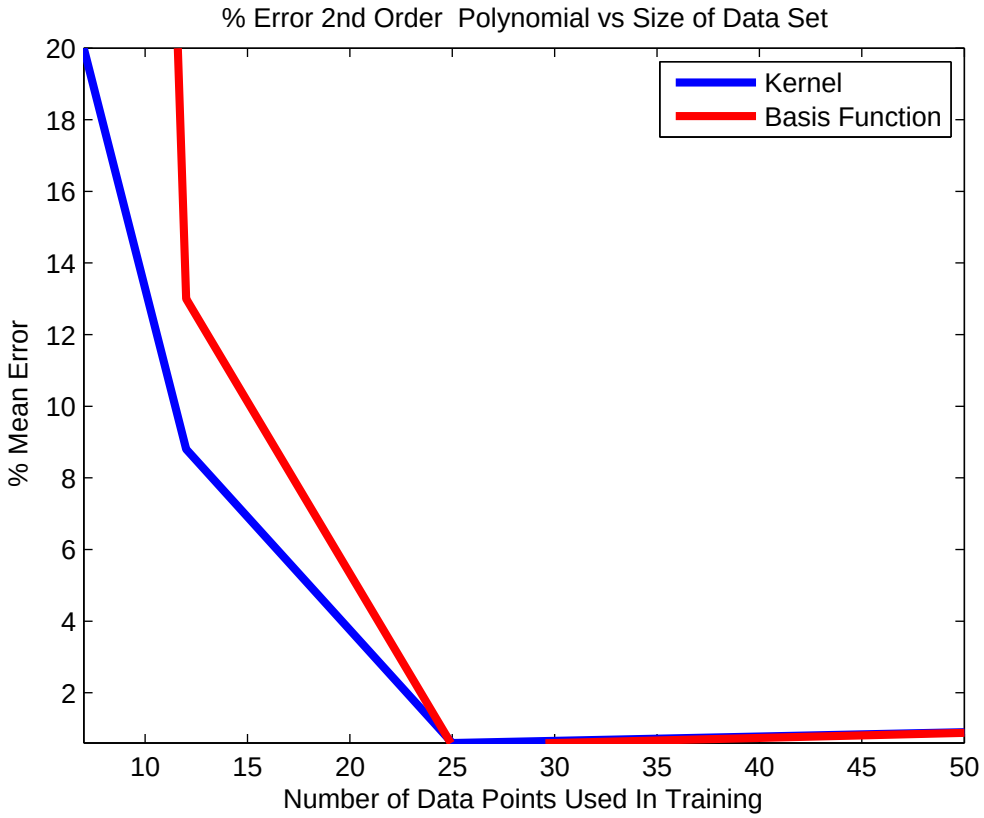


Figure 5.11: y-axis represents the average RSE for a 2nd order polynomial and the corresponding basis function using simulated data, x-axis represents the number of samples used in training

To better visualize the process, a one-dimensional case is generated for $d = 2$ shown in Fig. 5.12. The red line on the top of the figure represents the total output and the green and blue represent the output of the two experts. The bottom of the figure shows the output of the Softmax functions. As the softmax functions decay, so does the contribution to the output of that expert.

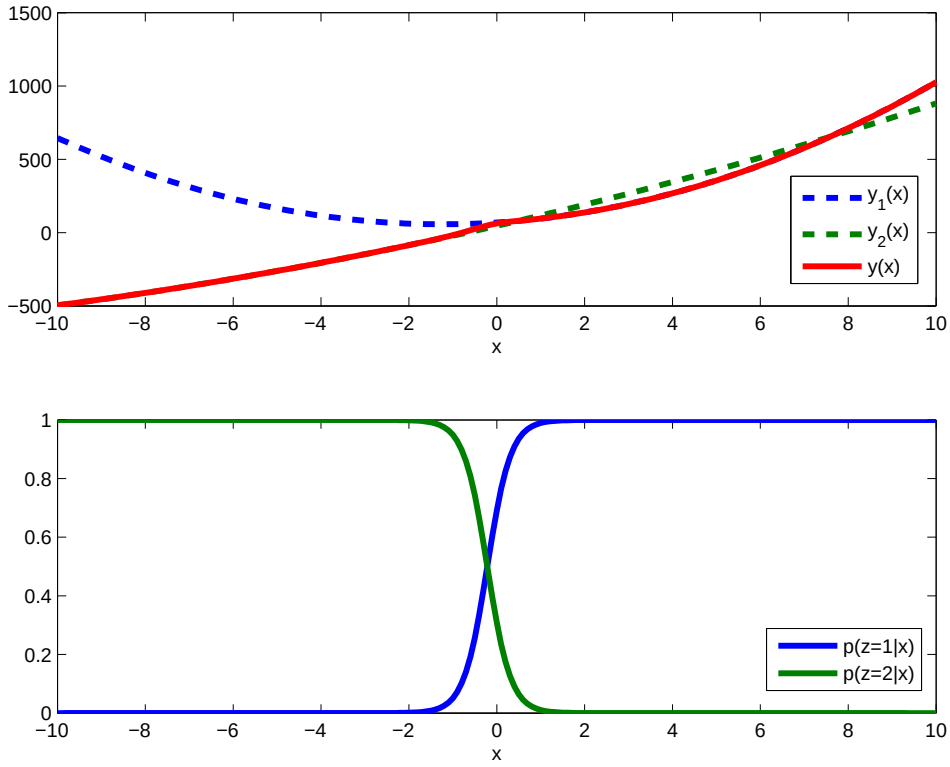


Figure 5.12: Top: Total output and output of each expert, Bottom: Value of Softmax functions

5.9.2 Results on LIRIS: Kernel-Based Mixture of Experts

Table. 5.3 compares the classic mixture of experts to the novel mixture of experts. Each row corresponds to the kernel or basis function used and the columns correspond to the prediction. The Quadratic function can be generated using a kernel, but, as the results are comparable using a basis function, we include them in the realm of classic mixture of experts. The novel method uses the RBF kernel as that kernel is impossible to implement using basis functions. Examining the results, it is evident that the novel method using the RBF kernel performs well. The massive error in the other methods is due to the fact that the other methods functions are not bounded. This is demonstrated in the top of Fig. 5.13, with a polynomial kernel corresponding to a training sample of one. We see that as the input changes the kernel value changes and the values increase drastically. This is not a problem in classification, as large values simply correspond to values that

are far away from the decision boundary. In regression however, this is problematic, as these large values may be influenced by outliers. As a result, some values will have extremely large positive or negative values. This is not a problem for the RBF as they are maximal only for values close to the specified samples as shown in the bottom of Fig. 5.13. Furthermore, there is an RBF tuning parameter that can control over-fitting. The method performs comparable to other methods shown in Table. 5.3. One important note is that this model is not prone to over fitting, unlike other neural networks. Most neural network packages, (including Matlab used in this course), use complicated validation procedures to determine network parameters such as number of layers and regulation parameters. The novel method has the same performance but only requires a two dimensional grid search for the free parameters and does not have the disadvantage of being a black box.

Table 5.3: Average RSE for classic mixture of experts, method using novel mixture of experts and kernels

Method	Valence	Arousal
Linear	9.24%	33%
Quadratic	6.4%	38%
Novel **RBF**	**4.20%**	**12.34%**
Neural Network	4.1632%	12.1757%

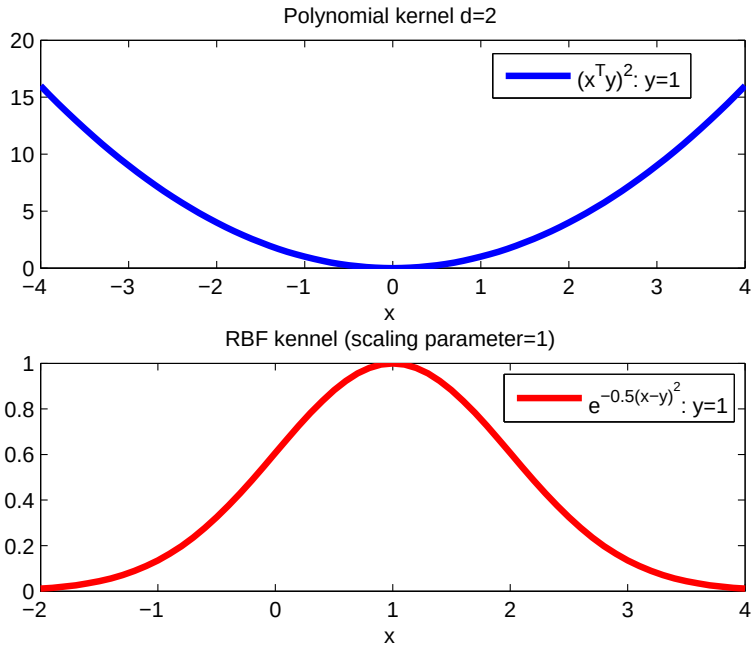


Figure 5.13: 1-D plot of different kernels Top: Polynomial kernel Bottom: RBF kernel.

5.10 Conclusion

This section develops DPHMM and a kernel-based mixture of experts models for linear regression to estimate emotional responses. The contributions for the DPHMM include a new probabilistic method to predict the value of sequential data, a novel dynamic programming algorithm, a novel use of the EM algorithm, and the first experiments performed in ATC in sports videos. The advantages of the models are that state training is unsupervised and the method outperforms other regression methods in arousal analysis. The drawbacks of the models are that the data must exhibit sequential patterns and determination of the ATC is more computationally intensive than most of the methods it was tested against. In addition, we formulate a novel mixture of experts models for linear regression so that kernel functions can be used. This avoids the problem encountered when dealing with a large feature space that can lead to serious computational difficulties. Other advantages of the model include the ability to take advantage of all the work related to kernels, a closed-form solution for maximization, as well as maintaining all the advantages of a linear expert. The model is verified and tested with simulated data, and it is also found that the model

has overall better performance than ME on simulated data. The method also performs better than starred ME on the LIRIS data set, and, comparable to other methods. Kernels used included linear, polynomial, and radial basis functions.

Chapter 6

A Textural Based Hidden Markov Model for Animation Genre Discrimination

6.1 Introduction

Different types of animation are used to gain children's attention [24, 25, 26], and imaginary characters mixed with live action improve attention [80]. As a result, we develop a method to classify different types of animated content that performs better than more complex general methods for video genre classification. It is found that this method, using a few specialized features, has better performance than generic methods. This is the first such research done in animation genre categorization. The animation genres include hand drawn animation (HDA), computer animation (CA), and stop motion animation (SMA). The method also works for black and white animation unlike [82]. The system can be used with a standard video genre discrimination system as shown in the diagram in Fig. 6.2. After video genre classification has been performed, the system can be used to classify the animation into subcategories.

Animation genre information does not appear available in meta-data. Consider Fig. 6.1, which shows the percentage of occurrences of the term stop motion animation, claymation, animation, hand drawn animation or cartoon in videos used in our data set. The titles, tags, comments and related videos in the list were used in a query. The term cartoon occurs the most with under 20%, the rest occur with much less frequency as shown in Fig. 6.1.

In animation, colour and edge features are similar for different animation genres and CA and

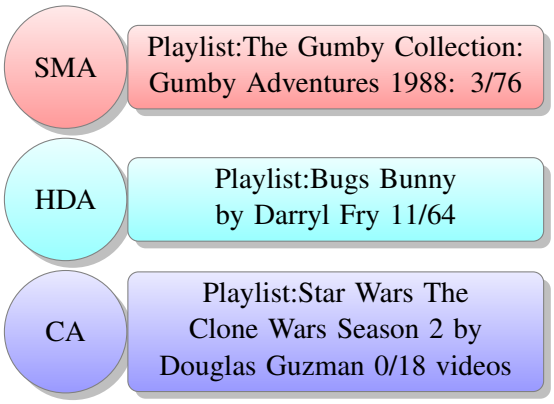


Figure 6.1: Occurrence of tags relating to different animation genres for several videos in the dataset.

hand drawn animation have similar motion. The characteristics in different animation genres appear much more similar, therefore a method is developed based on modeling the temporal characteristics of the texture. This is referred to as a temporal texture model [150]. The work models the texture properties by using gray level co-occurrence matrix (GLCM) and HMM (GLCMHMM).

The contributions include a temporal texture model based on GLCM. There has been research in animation genre categorization that uses colour [82], but the method developed here does not use colour allowing it to classify older black and white content. It is found that the GLCMHMM has over 85.71% accuracy and outperforms block intensity comparison code (BICC) [56] in this specific classification task using both HMM and SVM.

6.2 Problem Formulation

In this section, we estimate the parameters of each genre using a training set, then classify a video based on the parameters that have the highest likelihood as shown in equation 6.1:

$$\bar{\theta} = \arg\max_{\theta_{genre}} \{P(O|\theta_{genre})\}. \tag{6.1}$$

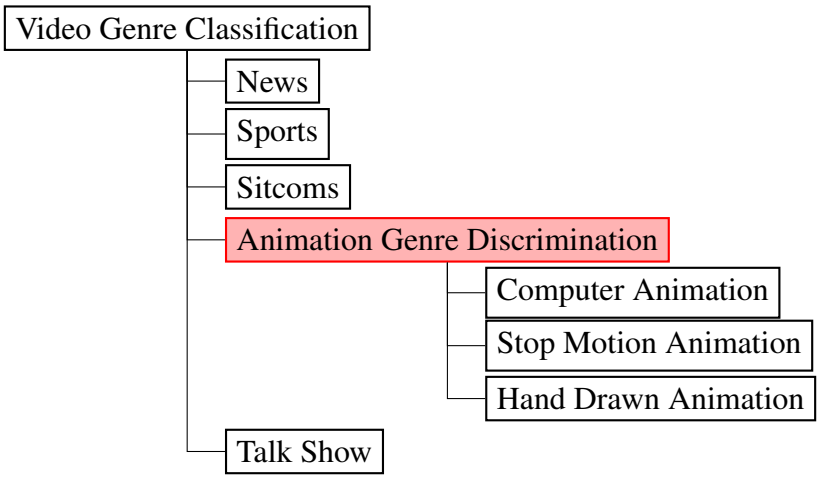


Figure 6.2: Tree representing a system to classify animation genre, first video genre classification is performed, then animation genre discrimination is performed. With stop motion animation (SMA) and hand drawn animation (HDA).

Figure 6.3: Example of one frame of hand drawn animation.

6.2.1 Animation Genres

Hand-drawn animation is created by individual frames first drawn on paper. To create the illusion of movement, each drawing differs slightly from the one before it. The animators' drawings are then copied onto acetate sheets or cells, which are filled in with paints in assigned colours on the side opposite the line drawings. The completed character cells are photographed individually onto motion picture film against a painted background using a rostrum camera; an example of hand drawn animation is shown in Fig. 6.3. Modern methods of hand drawn animation use computers to assist but the general appearance is the same [151].

CA, a three dimensional (3D) image model, is constructed out of polygons and projected into a two-dimensional image. In more recent CA methods, coarseness caused by the polygons is

Figure 6.4: Example of one frame of CA.

Figure 6.5: Example of one frame of Stop motion animation.

smoothed using a number of methods. Like hand drawn animation, images are characterized by geometric homogenous shapes. However, unlike hand drawn animation, these images appear 3D due to factors such as occlusions, lighting interactions, shading, shadows, and texture mapping [151]. An example of CA is shown in Fig. 6.4.

Stop motion animation (SMA) is created by manipulating sculptures and photographing the successive changes one frame of film at a time, in order to create the illusion of movement [151]. An example of stop motion animation is shown in Fig. 6.5.

6.3 Feature Space

Perceptually, it is simple to distinguish between the different genres. Texture features seem ideal because spatially, the global texture attributes appear the same within each genre. To quantify the textural differences a GLCM is used.

The GLCM provides a second-order method for generating texture features in gray scale im-

ages [152]. There are many other methods to analyze texture [153], but GLCM 's have several advantages. They are relatively computationally inexpensive, simple to implement, and the features are particularly well-suited for this application as discussed below.

6.3.1 Gray Level Co-occurrence Matrix

Let $Im(\ddot{x}, \ddot{y})$ be the gray level intensity array image with dimensions $\ddot{X}$ by $\ddot{Y}$. Every element of a GLCM represents the number of occurrences of gray levels g_x and g_y with spatial and displacement: $\Delta\ddot{x} \in Z$ and $\Delta\ddot{y} \in Z$. If the GLCM is normalized it can be interpreted as a probability density function (PDF) representing the probability that a pixel of $Im(\cdot)$ has the same intensity displaced by $[\Delta\ddot{x}, \Delta\ddot{y}]$, the PDF is given by:

$$P(g_x, g_y) = \sum_{\ddot{y}=1}^{\ddot{Y}-\Delta y} \sum_{\ddot{x}=1}^{\ddot{X}-\Delta\ddot{x}} I[Im(\ddot{x}, \ddot{y}) - g_y, Im(\ddot{x}+\Delta\ddot{x}, \ddot{y}+\Delta\ddot{y}) - g_y]/(\ddot{X}-\Delta\ddot{x})(\ddot{Y}-\Delta\ddot{y}) \tag{6.2}$$

Consider the following example: The hand drawn animation in Fig. 6.3, the image is composed of homogeneous colours and as a result, there will only be a few dominant gray tones, therefore $P(g_x, g_y)$ will have a few entries of large magnitude. Now consider the CA image shown in Fig. 6.4. The image has a similar appearance but factors such as shading, lighting, and shadow will lead to a larger number of smaller entries. A symmetric $P(g_x, g_y)$ can also be calculated by combining negative and positive values for displacement. The next section will discuss selecting parameters together with some of the statistical measurers extracted from the GLCM.

6.3.2 Parameters of Gray Level Co-occurrence Matrix

There are three fundamental parameters that must be determined in selecting a GLCM: Quantization levels (QL) of the image and the displacement and orientation. Before a GLCM is calculated, the image is usually re-quantized, the standard of 16 QL is used. Many factors, such as image

resolution and noise, determine the best QL for optimum classification accuracy [154]. 16 QL, 24 QL and 32 QL are tested, and literature suggests that over 32 QL is redundant [154]. To allow for easy comparison, displacement is set to one, and horizontal and vertical displacement are used. For a more statistically based method of selecting the parameters of a GLCM see [155].

6.3.3 Textural Features Extracted from Gray Level Co-occurrence Matrix

This paper uses 22 features. One of the equations is given below with the expected results, while the rest are given in Table. 6.7 with their corresponding reference.

Contrast measures the local intensity variation and is given by equation 6.3. One would expect stop motion animation to have a large contrast because it does not have the smooth appearance of CA and the homogenous colours of hand drawn animation.

$$f_2 = (\sum_{g_x} \sum_{g_y} |g_x - g_y|^2 (p(g_x, g_y))) \tag{6.3}$$

The rest of the measurements are given in Table. 6.7. For every frame t a GLCM is obtained and the GLCM features are extracted into t feature vector of 22 dimensions:

$$\mathbf{o}_t = [\ f_{1,t}\ f_{2,t}\ ....\ f_{d-1,t}\ f_{22,t}]^{\mathbf{T}} \tag{6.4}$$

Table 6.1: GLCM features used with corresponding reference, the following acronyms are used Inverse Difference (ID) Inverse Measure (IM).

Autocorrelation [156]	Correlation [152]	Cluster Shade (CS)[156]	Cluster Prominence (CP)[156]
Dissimilarity[156]	Homogeneity [152]	Homogeneity[156]	Maximum Probability (MP) [156]
Variance [152]	Sum Average (SA)[152]	Sum Variance (SV)	Sum Entropy (SE) [152]
Difference Variance (DV)[152]	Difference Entropy (DI)[152]	IM of Correlation 1 (IMC2)[152]	ID normalized (INN)[154]
ID Homogeneity (IDH) [152]	IDH Normalized (IDHN) [154]	IM of Correlation 2 (IMC2)[152]	ID moment normalized [154]

6.4 Textural Hidden Markov Model

An early example of classifying texture with motion includes [150], which used Autoregressive models. In this section we formulate a novel temporal texture model using HMM. Once the textural features have been extracted, the HMM will capture temporal relationships between frames. For example, in stop motion animation each character must be manipulated manually. There is no camera and background motion, which means the textural features extracted remain similar from frame to frame. In CA there is an abundance of change caused from occlusions, lighting interactions, shading, shadows, texture mapping, camera motion and blurring.

For this section a Gaussian mixture model (GMM) with K_{mix} mixtures is used as the observation distribution and is given in equation 6.5.

$$b_i(\mathbf{o}_t) = \sum_{k=1}^{K_{mix}} w_{k,i} \mathcal{N}\left(\mathbf{o}_t | \mu_{ki}, \Sigma_{ki}\right) \tag{6.5}$$

This distribution represents observations that are inherent in each state. The parameter $\{w_k\}$ must satisfy constraints 6.6 and 6.7.

$$\sum_{k=1}^{K_{mix}} w_k = 1 \tag{6.6}$$

$$0 \leq w_k \leq 1 \tag{6.7}$$

Equation 6.5 represents the maximum likelihood estimate (MLE) of the probability of observing $\mathbf{o}_t$ and being in state i at some time t. The probability of observing a sequence O can be obtained by using equation 6.8.

For each genre the parameters θ_{genre} are obtained using Baum-Welch re-estimation (BWRE) [55]. Once the parameters have been obtained for each genre, we calculate the probability of observing that particular state sequence using:

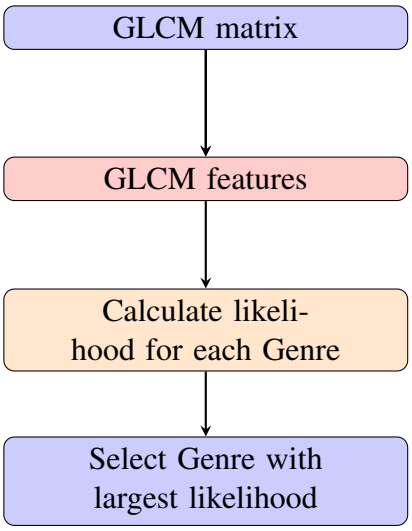


Figure 6.6: The GLCM matrix and GLCM features are determined. Then the likelihood for each genre is calculated. The sequence's genre is determined by selecting the HMM with the largest likelihood.

$$P(O|\theta_{genre}) = \sum_{(q_1,q_2,..,q_T)} = \pi_{q_1} b_{q_1}(\mathbf{o}_1) a_{q_1 q_2} b_{q_1}(\mathbf{o}_2) ... a_{q_{T-1} q_T} b_{q_T}(\mathbf{o}_T). \tag{6.8}$$

6.5 Block Diagram: Textural Based HMM

The final block diagram is shown in Fig. 6.6. The GLCM matrix and GLCM features are determined. This process is represented with a purple and pink block respectively. Then the likelihood for each genre is calculated and denoted with the orange block. Finally the sequence's genre is determined by selecting the HMM with the largest likelihood represented with the purple block at the bottom.

6.6 Experimental Procedure

The test set consists of hand drawn animation obtained from several popular TV shows. CA is taken from short clips of Pixar movies and stop motion animation clips have been obtained from YouTube videos or TV shows. No two videos come from the same movie or clip. Some videos

Table 6.2: Percentage accuracy of different methods

Offset ($\Delta\ddot{x}, \Delta\ddot{y}$)	(1,0)	(0,1)	(1,1)
Accuracy	83.33%	78.57%	SCM

Table 6.3: Percentage accuracy of different QL ($\Delta\ddot{x} = 1, \Delta\ddot{y}$=0)

QL	16	24	32
Accuracy	83.33%	73.81%	SCM

come from the same franchise, for example Toy Story 1, 2 and 3 but all were kept in the same training or testing fold. Seven of the videos are used for training and seven of the videos are used for testing. The clips range in length from as short as 30 seconds to as long as 4 minutes. The quality, resolution, format, and aspect ratio also vary depending on the source of the video. The videos are all converted to Windows media player format using a Real video player converter and all processing has been done in Matlab. A total of 7 videos have been used for training and 14 for testing or three fold cross-validation.

6.7 Results

In this work, 42 videos have been tested, 14 from each genre. For determining the optimal number of states and comparing the number of Gaussian mixtures, the observation distribution is set to one. The optimum number of states is 6, as more than 7 states led to singular covariance matrices determined empirically.

As previously stated, different parameters for the GLCM have also been tested, some leading to singular covariance matrices (SCM) in the HMM observation distribution. The results for different offset parameters and different QL are shown in Table 6.2 and Table 6.3.

It was found that GLCMFBHMM outperforms the BICC method for a 22 Dimension (22 D) feature vector and for the optimum number of a 100 Dimension (100 D) feature vector as determined by [56]. The results are shown in table 6.4 also BICC using HMM and in Table 6.5 the classifier used was SVM.

Table 6.4: Percentage accuracy of GLCMFBHMM vs HMM and BICC

Method	BICC 22 D (HMM)	BICC 100 D (HMM)	GLCMFBHMM
Accuracy	66.67%	73.33%	83.33%

Table 6.5: Percentage accuracy of GLCMFBSVM vs SVM and BICC using (RBF) kernel

Method	BICC 22 D (SVM)	BICC 100 D (SVM)	GLCMFBSVM
Accuracy	62.67%	71.33 %	74.33%

The GLCMFBHMM performed best, in general the HMM models outperformed the SVM for the same feature set, this is most likely due to the fact that SVM works better with a much larger number of features [1].

The confusion matrix is also used to evaluate the algorithm. It is evident from table 6.6 that even with almost five times more features, BICC with HMM did not have the classifying ability of GLCMFBHMM.

Once the optimum number of states is determined, the optimum number of mixtures is determined to be three. The increasing error for more than three states is most likely due to over-fitting of the training data. The confusion matrix for the optimum number of states is given in table 6.7. The optimum accuracy is 85.71% with six states performing best. The sample standard deviation was 3.13% making the normal approximation of the accuracy approximately between 80% to 90% with 95% confidence.

Comparing Table 6.7 with Table 6.6, it is evident that changing the number of mixtures does not have any substantial improvement. Therefore, the data has well concentrated decision regions. This is also beneficial in training, in that using one mixture requires less computation and is less sensitive to initialization.

6.8 Conclusion

This chapter develops the novel GLCMFBHMM as a method to categorize hand-drawn animation, CA and stop motion animation. The features extracted from the GLCM are used to quantify the

Table 6.6: Confusion matrix for different methods, with the same ordering as table 6.4

Genres	Drawn	Stop Motion	Computer
Drawn	(7,9,10)	(0,4,0)	(8,1,4)
Stop Motion	(0,3,1)	(8,11,11)	(6,0,2)
Computer	(0,0,0)	(1,2,0)	(13,12,14)

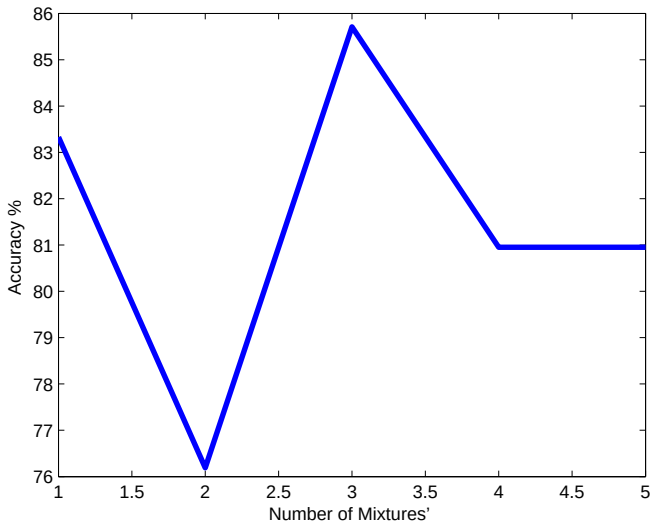


Figure 6.7: Number of mixtures vs accuracy.

different textures; then an HMM is used to classify each animation genre. The contributions include the first use of texture features to classify animation genre that can be used to classify colour and non-colour animation, the use of HMM in temporal texture modelling, and a temporal texture model based on GLCM.

It is found that the GLCMFBHMM functions best with GCLM's with 16 QL and horizontal offsets. The GLCMFBHMM has 85.71% accuracy and outperforms BICC by 16.66% for the same number of parameters and dimensions.

The drawbacks of the method relate to its requirement of multiple frames, and the HMM requires huge volumes of training data in high dimensional space. SCM suggests some features have low variances or have bad scaling. Since the video data exhibits time varying patterns, various experiments must be conducted to investigate the performance of each of the features and redundancy

Table 6.7: Confusion matrix for a six state, three mixture HMM.

Genres	Drawn	Stop Motion	Computer
Drawn	11	0	3
Stop Motion	1	11	2
Computer	0	0	14

of features.

Interesting avenues of research to follow include the use of textural based features in the standard video genre classification paradigm, the use of different textural based features, using textural based features, and more advanced classification methods [157],[158],[159].

Chapter 7

Conclusion

This thesis introduces the concept of cognitive content of a video sequence. The work focuses on modelling and classifying the interaction between an individual's cognition and the audio and visual components of video content. Chapter 1 introduces the topic and positions it in the context of current multimedia and scientific research. Chapter 2 then formulates the problem in terms of recursive risk minimization and gives important background information with respect to statistical models.

Chapter 3 introduces the novel concept of positive developmental video classification for children. In addition to the novel research topic, we collect a set of videos that have been deemed as having a negative or positive impact on child cognition from a literature review. A novel model validation technique is developed, and several new features and experiments are conducted.

Chapter 3 also introduces automatic age-based classification. As a novel research topic, we introduce several novel high-level audio features related to the cognitive capacity of children. These novel features gauge the cognitive ability of the intended audience by quantifying the structure of the language. These novel features include syllable rate, word rate, language complexity and noise jumps. The feature extraction methods are also novel in that we count the number of syllables and words using relatively computationally inexpensive signal processing techniques that forgo complex speech recognition.

Given the accuracy of affective features, the relationship between emotions, cognition and the impact of arousal, we focus on affective ranking in this chapter. The main contributions of this

chapter include the development of a method to rank sequences using their affective content without the granularity problem. In addition, the linking transform is developed to incorporate prior knowledge into the cluster assignments. Furthermore, the work compares the accuracy of several regression methods on the LIRIS database and performs regularization and variable selection via the Elastic Net and Lasso methods.

Chapter 5 develops several state based models to map features onto the valence and/or arousal plane. The methods include dynamic prediction hidden Markov models for arousal time curve estimation in sports videos and Kernel-based mixtures of experts for linear regression. The dynamic prediction hidden Markov model determines the arousal time curve by selecting a state sequence that maximizes the joint probability density function between the states and the arousal time curve. We derive the parameters using the expected maximization algorithm. Experiments are performed on several types of sports videos including golf, bowling, darts and tennis. Test measures include squared residual error and criteria derived from psychology. The experimental results show that the novel method performs better in estimating the arousal time curve than state-of-the art linear regression methods on most of the tested sports videos. The Kernel-based mixture of experts is also developed, with the method outperforming other mixture of expert models and exhibiting comparable performance to other methods for regression on the modelling using the LIRIS database.

Due to the use of animation as a means of obtaining children's attention, chapter 6 introduces a method to automatically categorize different animation genres in a video database made for children. The method is based on statistically modelling the temporal texture attributes of the video sequence and unlike other methods can be used for black and white content.

Future Work

Most models do not take into account how duration impacts cognition, as they assume each sample of the data to be independent. Thus models that do not make the independence assumption between data samples in the video sequence should be explored.

Examining how features can be used to better predict how well an individual will understand and absorb content would also be useful. This could be done with speech rate, using the rela-

tionship between valence, arousal and attention or a myriad of other ways. In future work feature comparisons of high-level, mid-level and low-level features can also be incorporated.

As the scientific literature states that the impact of video is different on adults and children, the video impact on adults should also be explored. This would require joint research in the scientific and engineering communities.

Although this thesis finds that in many cases low level features could be used for determining the cognitive impact of video content, more semantic features such as bag of words and visual bag of words could also be used to determine if specific instances of words or objects have an impact on cognition. Other methods such as deep networks could also be applied but these methods require much more data [40, 41]. Deep networks could incorporate semantic information and improve general classification results on the video sequence. In addition, deep networks could be incorporated in the audio component of classification which has been found to be important in the cognitive classification, improving results by encoding complex semantic information from the audio component.

Physiological data should also be included in the training data as it is a more robust means of determining the impact in cognition. Another interesting avenue would be to predict the impact on cognitive content using diagnostics taken before and after exposure to content.

Appendix: Proof Linking Transformation Satisfies Mercer's theorem if gram matrix is positive semidefinite

Theorem: Let $\hat{\mathbf{K}}_L$ be the $N \times N$ transformed kernel matrix, if $\mathbf{K}$ satisfies Mercer's theorem then $\hat{\mathbf{K}}_L$ satisfies Mercer's theorem.

Proof: If $\mathbf{K}$ is a valid kernel matrix then Mercer's theorem states that $\mathbf{K}$ is positive semi-definite: $\boldsymbol{\alpha}^T\mathbf{K}\boldsymbol{\alpha} \geq 0$. Where $\boldsymbol{\alpha} \in \Re^N$. Then we must show that $\boldsymbol{\beta}^T\hat{\mathbf{K}}_L\boldsymbol{\beta} \geq 0$ hold such that $\boldsymbol{\beta} \in \Re^N$. The proof is completed by induction.

Base Case:

$$\boldsymbol{\alpha}^T\hat{\mathbf{K}}_1\boldsymbol{\alpha} = \boldsymbol{\alpha}^T\mathbf{F}_1\mathbf{K}\mathbf{F}_1\boldsymbol{\alpha} = \boldsymbol{\beta}^T\mathbf{K}\boldsymbol{\beta} \geq 0$$

Where $\mathbf{F}_1\boldsymbol{\alpha} = \boldsymbol{\beta}$ and $\boldsymbol{\beta} \in \Re^N$. This is because $\mathbf{F}_1$ is full rank. Similarly $\boldsymbol{\alpha}^T\mathbf{F}_1 = \boldsymbol{\beta}^T$. Therefore as $\mathbf{K}$ is positive semi-definite then $\boldsymbol{\alpha}^T\hat{\mathbf{K}}_1\boldsymbol{\alpha} \geq 0$ and $\hat{\mathbf{K}}_1$ is positive semi-definite and satisfies Mercer's theorem.

Induction step:

$$\boldsymbol{\alpha}^T\hat{\mathbf{K}}_L\boldsymbol{\alpha} = \boldsymbol{\alpha}^T\mathbf{F}_L\hat{\mathbf{K}}_{L-1}\mathbf{F}_L\boldsymbol{\alpha} = \boldsymbol{\beta}^T\hat{\mathbf{K}}_{L-1}\boldsymbol{\beta} \geq 0$$

As above $\boldsymbol{\alpha}^T\mathbf{F}_L = \boldsymbol{\beta}^T$ and $\mathbf{F}_L\boldsymbol{\alpha} = \boldsymbol{\beta}$. As $\hat{\mathbf{K}}_{\mathbf{L-1}}$ as positive semi-definite and $\boldsymbol{\beta} \in \Re^N$ then $\boldsymbol{\beta}^T\hat{\mathbf{K}}_{L-1}\boldsymbol{\beta} \geq 0$. Therefore $\hat{\mathbf{K}}_L$ satisfies Mercer's theorem.

Bibliography

[1] H.K. Ekenel and R. Stiefelhagen, "Content-based video genre classification using multiple cues," in *Proceedings of the 3rd international workshop on Automated information extraction in media production*. ACM, 2010, pp. 21–26.

[2] A. Hanjalic and L-Q. Xu, "Affective video content representation and modeling," *Multimedia, IEEE Transactions on*, vol. 7, no. 1, pp. 143–154, 2005.

[3] Nielsen, "MS Windows NT kernel description," 2009.

[4] V.J. Rideout, U.G. Foehr, and D.F Roberts, "Generation m : Media in the lives of 8-to 18-year-olds.," *Henry J. Kaiser Family Foundation*, 2010.

[5] S.C. Burke and S.L. Snyder, "Youtube: an innovative learning resource for college health education courses.," *International Electronic Journal of Health Education*, vol. 11, pp. 39–46, 2008.

[6] J. Copley, "Audio and video podcasts of lectures for campus-based students: production and evaluation of student use," *Innovations in education and teaching international*, vol. 44, no. 4, pp. 387–399, 2007.

[7] A. Clifton and C. Mann, "Can youtube enhance student nurse learning?," *Nurse education today*, vol. 31, no. 4, pp. 311–313, 2011.

[8] J. Agazio and K.M. Buckley, "An untapped resource: Using youtube in nursing education," *Nurse educator*, vol. 34, no. 1, pp. 23–28, 2009.

[9] L.J. Shrum, "Assessing the social influence of television a social cognition perspective on cultivation effects," *Communication Research*, vol. 22, no. 4, pp. 402–429, 1995.

[10] A. Lang and Z. Wang, "Cognition and emotion in TV message processing: How valence, arousing content, structural complexity, and information density affect the availability of cognitive resources," *Media Psychology*, vol. 10, no. 3, pp. 317–338, 2007.

[11] A. Lang, "The limited capacity model of mediated message processing," *Theorizing communication, readings across traditions*, 2007.

[12] A. Lang and B. Reeves, "Negative video as structure: Emotion, attention, capacity, and memory," *Journal of Broadcasting & Electronic Media*, vol. 40, no. 4, pp. 460–477, 1996.

[13] A. Lang and K. Kawahara, "The effects of production pacing and arousing content on the information processing of television messages," *Journal of Broadcasting & Electronic Media*, vol. 43, no. 4, pp. 451–475, 1999.

[14] A. Lang, K. Dhillon, and Q. Dong, "The effects of emotional arousal and valence on television viewers cognitive capacity and memory," *Journal of Broadcasting & Electronic Media*, vol. 39, no. 3, pp. 313–327, 1995.

[15] A. Maass, K.M. Klöpper, F. Michel, and A. Lohaus, "Does media use have a short-term impact on cognitive performance? a study of television viewing and video gaming.," *Journal Of Media Psychology: Theories, Methods, And Applications*, vol. 23, no. 2, pp. 65, 2011.

[16] A. Maass and A. Lohaus, "Effects of violent and non-violent computer game content on memory performance in adolescents," *European journal of psychology of education*, vol. 26, no. 3, pp. 339–353, 2011.

[17] R. Dietz and A. Lang, "Affective agents: Effects of agent affect on arousal, attention, liking and learning," in *Proceedings of the Third International Cognitive Technology Conference, San Francisco*, 1999.

[18] A. Lillard and S. Peterson, "The immediate impact of different types of television on young children's executive function," *Pediatrics*, vol. 128, no. 4, pp. 644–649, 2011.

[19] D.A. Christakis, J.S.B. Ramirez, and J.M. Ramirez, "Overstimulation of newborn mice leads to behavioral differences and deficits in cognitive performance," *Scientific reports*, vol. 2, 2012.

[20] D.A. Christakis, "The effects of fast-paced cartoons," *Pediatrics*, vol. 128, no. 4, pp. 772–774, 2011.

[21] D.R. Anderson, S.R. Levin, and E.P. Lorch, "The effects of TV program pacing on the behavior of preschool children," *Educational Technology Research and Development*, vol. 25, no. 2, pp. 159–166, 1977.

[22] H. Kirkorian, E. Wartella, and D. Anderson, "Media and young children's learning," *The Future of Children*, vol. 18, no. 1, pp. 39–61, 2008.

[23] S.M. Fisch and R.T. Truglio, *G is for growing: Thirty years of research on children and Sesame Street*, Routledge, 2014.

[24] C. Christakis and A. Dimitri, "The effects of infant media usage: what do we know and what should we learn," *Acta Paediatrica*, vol. 98, no. 1, pp. 8–16, 2009.

[25] S. Calvert, A. Huston, B. Watkins, and J. Wright, "Children's processing of television: The informative functions of formal features," *Childrens understanding of television: Research on attention and comprehension*, pp. 35–68, 1983.

[26] S.A. Goodrich, T.A. Pempek, and S.L. Calvert, "Formal production features of infant and toddler dvds," *Archives of Pediatrics & adolescent medicine*, vol. 163, no. 12, pp. 1151–1156, 2009.

[27] L.G. Naigles and E.T. Kako, "First contact in verb acquisition: Defining a role for syntax," *Child development*, vol. 64, no. 6, pp. 1665–1687, 1993.

[28] M.E. Schmidt and E.A. Vandewater, "Media and attention, cognition, and school achievement," *The Future of children*, vol. 18, no. 1, pp. 63–85, 2008.

[29] A.B. Tucker, *Computer science handbook*, CRC press, 2004.

[30] S.A. Dayhoff, *Diagonally-parked in a parallel universe: Working through social anxiety*, Effectiveness-Plus Publications, 2000.

[31] M.D. Lee and E-J. Wagenmakers, *Bayesian cognitive modeling: A practical course*, Cambridge University Press, 2014.

[32] J. Santarcangelo and X-P. Zhang, "Arousal content representation of sports videos using dynamic prediction hidden Markov models," in *Signal and Information Processing (GlobalSIP), 2014 IEEE Global Conference on*. IEEE, 2014, pp. 1049–1053.

[33] J. Santarcangelo and X-P. Zhang, "Kernel-based mixture of experts models for linear regression," in *Circuits and Systems (ISCAS), 2015 IEEE International Symposium on*. IEEE, 2015.

[34] Y. Baveye, E. Dellandrea, C. Chamaret, and L. Chen, "Liris-accede: A video database for affective content analysis," *Affective Computing, IEEE Transactions on*, vol. 6, no. 1, pp. 43–55, 2015.

[35] W. Hu, N. Xie, L. Li, X. Zeng, and S. Maybank, "A survey on visual content-based video indexing and retrieval," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 41, no. 6, pp. 797–819, 2011.

[36] D. Brezeale and D. Cook, "Automatic video classification: A survey of the literature," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 38, no. 3, pp. 416–430, 2008.

[37] B. Ionescu, I. Mironica, K. Seyerlehner, P. Knees, J. Schlüter, M. Schedl, H. Cucu, A. Buzo, and P. Lambert, "Arf@ mediaeval 2012: Multimodal video classification.," in *MediaEval*, 2012.

[38] S.W. Smoliar and H.J. Zhang, "Content-based video indexing and retrieval," *IEEE multimedia*, , no. 2, pp. 62–72, 1994.

[39] J.R. Zhang, Y. Song, and T. Leung, "Improving video classification via youtube video co-watch data," in *Proceedings of the 2011 ACM workshop on Social and behavioural networked media access*. ACM, 2011, pp. 21–26.

[40] J. Yue-Hei, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4694–4702.

[41] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[42] Y-L. Boureau, J. Ponce, and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition," pp. 111–118, 2010.

[43] C. Hulme, S. Maughan, and G. Brown, "Memory for familiar and unfamiliar words: Evidence for a long-term memory contribution to short-term memory span," *Journal of Memory and Language*, vol. 30, no. 6, pp. 685–701, 1991.

[44] Y. Wu, B.L. Tseng, and J.R. Smith, "Ontology-based multi-classification learning for video concept detection," in *Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on*. IEEE, 2004, vol. 2, pp. 1003–1006.

[45] J.S. Boreczky and L.A. Rowe, "Comparison of video shot boundary detection techniques," *Journal of Electronic Imaging*, vol. 5, no. 2, pp. 122–128, 1996.

[46] W. Zhu, C. Toklu, and S-P. Liou, "Automatic news video segmentation and categorization based on closed-captioned text," in *null*. IEEE, 2001, p. 211.

[47] P. Xu, Y. Shi, and M.A. Larson, "Tud at mediaeval 2012 genre tagging task: Multi-modality video categorization with one-vs-all classifiers.," in *MediaEval*, 2012.

[48] J. Zhou and X-P. Zhang, "An ica mixture hidden Markov model for video content analysis," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 11, pp. 1576–1586, 2008.

[49] C.C. Cheng and C.T. Hsu, "Fusion of audio and motion information on HMM-based highlight extraction for baseball games," *Multimedia, IEEE Transactions on*, vol. 8, no. 3, pp. 585–599, 2006.

[50] V. Kobla, D. DeMenthon, and D.S. Doermann, "Identifying sports videos using replay, text, and camera motion features," in *Electronic Imaging*. International Society for Optics and Photonics, 1999, pp. 332–343.

[51] X. Gibert, H. Li, and D. Doermann, "Sports video classification using HMMs," in *icme*. IEEE, 2003, pp. 345–348.

[52] S. Vakkalanka, C. Krishna Mohan, R. Kumaraswamy, and B. Yegnanarayana, "Combining multiple evidence for video classification," in *Intelligent Sensing and Information Processing, 2005. Proceedings of 2005 International Conference on*. IEEE, 2005, pp. 187–192.

[53] G. Pass, R. Zabih, and J. Miller, "Comparing images using color coherence vectors," in *Proceedings of the fourth ACM international conference on Multimedia*. ACM, 1997, pp. 65–73.

[54] M.K. Geetha and S. Palanivel, "HMM based automatic video classification using static and dynamic features," in *Conference on Computational Intelligence and Multimedia Applications, International Conference on*.

[55] L.R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[56] G Kalaiselvi, S. Palanivel, and V. Ramalingam, "A novel block intensity comparison code for video classification and retrieval," *Expert Systems With Applications*, vol. 36, no. 3, pp. 6415–6420, 2009.

[57] A. Bandura, D. Ross, and S. Ross, "Imitation of film-mediated aggressive models.," *The Journal of Abnormal and Social Psychology*, vol. 66, no. 1, pp. 3, 1963.

[58] L. Friedrich and A. Stein, "Aggressive and prosocial television programs and the natural behavior of preschool children," *Monographs of the Society for Research in Child Development*, pp. 1–64, 1973.

[59] J. Nam, M. Alghoniemy, and A.H. Tewfik, "Audio-visual content-based violent scene characterization," in *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*. IEEE, 1998, vol. 1, pp. 353–357.

[60] S. Moncrieff, S. Venkatesh, and C. Dorai, "Horror film genre typing and scene labeling via audio analysis," in *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*. IEEE, 2003, vol. 2, pp. II–193.

[61] O. Deniz, I. Serrano, G. Bueno, and T.K. Kim, "Fast violence detection in video," in *The 9th International Conference on Computer Vision Theory and Applications (VISAPP)*, 2014.

[62] M. Montagnuolo and A. Messina, "TV genre classification using multimodal information and multilayer perceptrons," in *AI* IA 2007: Artificial Intelligence and Human-Oriented Computing*, pp. 730–741. Springer, 2007.

[63] M. Soleymani, G. Chanel, J.J.M. Kierkels, and T. Pun, "Affective ranking of movie scenes using physiological signals and content analysis," in *Proceedings of the 2nd ACM workshop on Multimedia semantics*. ACM, 2008, pp. 32–39.

[64] M. Soleymani, J.J.M. Kierkels, G. Chanel, and T. Pun, "A Bayesian framework for video affective representation," in *Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009. 3rd International Conference on*. IEEE, 2009, pp. 1–7.

[65] M. Soleymani, S. Koelstra, P. Sander, P. Ioannis, and T. Pun, "Continuous emotion detection in response to music videos," in *IEEE Automatic Face and Gesture Recognition, International Conference on*, March 21-March 25 2011, pp. 803–808.

[66] S. Koelstra, C. Muhl, M. Soleymani, J.S. Lee, A. Yazdani, T. Ebrahimi, T. Pun, A. Nijholt, and I. Patras, "Deap: A database for emotion analysis; using physiological signals," *Affective Computing, IEEE Transactions on*, vol. 3, no. 1, pp. 18–31, 2012.

[67] M. Xu and J. Jin, "Affective content analysis in comedy and horror videos by audio emotional event detection," in *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*. IEEE, 2005, pp. 4–pp.

[68] E. Acar and S. Albayrak, "Dai lab at mediaeval 2012 affect task: The detection of violent scenes using affective features.," in *MediaEval*. Citeseer, 2012.

[69] M. Wöllmer, F. Eyben, B. Schuller, E. Douglas-Cowie, and R. Cowie, "Data-driven clustering in emotional space for affect recognition using discriminatively trained lstm networks.," in *INTERSPEECH*, 2009, pp. 1595–1598.

[70] B. Schuller, R. Müller, F. Eyben, J. Gast, B. Hörnler, M. Wöllmer, G. Rigoll, A. Höthker, and H. Konosu, "Being bored? recognising natural interest by extensive audiovisual integration for real-life application," *Image and Vision Computing*, vol. 27, no. 12, pp. 1760–1774, 2009.

[71] E. Douglas-Cowie, R. Cowie, I. Sneddon, C. Cox, O. Lowry, M. Mcrorie, J-C. Martin, L. Devillers, S. Abrilian, and A. Batliner, "The humaine database: addressing the collection and annotation of naturalistic and induced emotional data," in *Affective computing and intelligent interaction*, pp. 488–500. Springer, 2007.

[72] A. Schaefer, F. Sanchez X. Nils, and P. Philippot, "Assessing the effectiveness of a large database of emotion-eliciting films: A new tool for emotion researchers," 2010, vol. 24, pp. 1153–1172, Taylor & Francis.

[73] M. Soleymani, J. Lichtenauer, T. Pun, and M. Pantic, "A multimodal database for affect recognition and implicit tagging," *Affective Computing, IEEE Transactions on*, vol. 3, no. 1, pp. 42–55, 2012.

[74] A.J. Blood and R.J. Zatorre, "Intensely pleasurable responses to music correlate with activity in brain regions implicated in reward and emotion," *Proceedings of the National Academy of Sciences*, vol. 98, no. 20, pp. 11818–11823, 2001.

[75] A. Lerch, *An introduction to audio content analysis: Applications in signal processing and music informatics*, John Wiley & Sons, 2012.

[76] Y-H. Yang, Y-C. Lin, Y-F. Su, and H.H. Chen, "A regression approach to music emotion recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, no. 2, pp. 448–457, 2008.

[77] Y. Wang and L. Guan, "Recognizing human emotional state from audiovisual signals*," *Multimedia, IEEE Transactions on*, vol. 10, no. 5, pp. 936–946, 2008.

[78] E. Scheirer and M. Slaney, "Construction and evaluation of a robust multifeature speech/music discriminator," in *International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 1997, vol. 2, pp. 1331–1334.

[79] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python*, " O'Reilly Media, Inc.", 2009.

[80] M. Gladwell, *The tipping point: How little things can make a big difference*, Little, Brown, 2006.

[81] A. Lang, "Involuntary attention and physiological arousal evoked by structural features and emotional content in TV commercials," *Communication Research*, vol. 17, no. 3, pp. 275–299, 1990.

[82] B. Ionescu, C. Vertan, P. Lambert, and A. Benoit, "A color-action perceptual approach to the classification of animated movies," in *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*. ACM, 2011, p. 10.

[83] V.N. Vapnik, "An overview of statistical learning theory," *Neural Networks, IEEE Transactions on*, vol. 10, no. 5, pp. 988–999, 1999.

[84] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 111–147, 1974.

[85] S. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge university press, 2004.

[86] C.M. Bishop, *Pattern recognition and machine learning*, springer, 2006.

[87] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

[88] P. Zhang, X. Wang, and P.X-K Song, "Clustering categorical data based on distance vectors," *Journal of the American Statistical Association*, vol. 101, no. 473, pp. 355–367, 2006.

[89] L. Xu, M.I. Jordan, and G.E. Hinton, "An alternative model for mixtures of experts," *Advances in neural information processing systems*, pp. 633–640, 1995.

[90] A. Hoerland and R. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, April 1970.

[91] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[92] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[93] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural computation*, vol. 4, no. 1, pp. 1–58, 1992.

[94] C. Saunders, A. Gammerman, and V. Vovk, "Ridge regression learning algorithm in dual variables," in *(ICML-1998) Proceedings of the 15th International Conference on Machine Learning*. Morgan Kaufmann, 1998, pp. 515–521.

[95] B. Schölkopf, R. Herbrich, and A.J. Smola, "A generalized representer theorem," in *Computational learning theory*. Springer, 2001, pp. 416–426.

[96] H.Q. Minh, P. Niyogi, and Y. Yao, "Mercers theorem, feature maps, and smoothing," in *Learning theory*, pp. 154–168. Springer, 2006.

[97] J. Platt et al., "Sequential minimal optimization: A fast algorithm for training support vector machines," 1998.

[98] H. Zha, X. He, C. Ding, M. Gu, and H.D. Simon, "Spectral relaxation for k-means clustering," in *Advances in neural information processing systems*, 2001, pp. 1057–1064.

[99] M. Welling, "Kernel k-means and spectral clustering," .

[100] D. Christakis, F. Zimmerman, D. DiGiuseppe, and C. McCarty, "Early television exposure and subsequent attentional problems in children," *Pediatrics*, vol. 113, no. 4, pp. 708–713, 2004.

[101] F. Zimmerman, D. Christakis, and A. Meltzoff, "Associations between media viewing and language development in children under age 2 years," *The Journal of pediatrics*, vol. 151, no. 4, pp. 364–368, 2007.

[102] A.M. Crawley, D.R. Anderson, A. Wilder, M. Williams, and A. Santomero, "Effects of repeated exposures to a single episode of the television program blue's clues on the viewing behaviors and comprehension of preschool children.," *Journal of Educational Psychology*, vol. 91, no. 4, pp. 630, 1999.

[103] K.L. Forge and S. Phemister, "The effect of prosocial cartoons on preschool children," *Child Study Journal*, vol. 17, no. 2, pp. 83–87, 1987.

[104] C.H. Huang and J.F. Wang, "Multi-weighted majority voting algorithm on support vector machine and its application," in *Proc. TENCON*. IEEE, 2009, pp. 1–4.

[105] P. Viola and M.J. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.

[106] A. Ramalingam and S. Krishnan, "Gaussian mixture modeling using short time fourier transform features for audio fingerprinting," in *IEEE International Conference on Multimedia and Expo*. IEEE, 2005, pp. 1146–1149.

[107] M. Bartsch and G. Wakefield, "Audio thumbnailing of popular music using chroma-based representations," *IEEE Transactions on Multimedia*, vol. 7, no. 1, pp. 96–104, 2005.

[108] D. Linebarger and D. Walker, "Infants and toddlers television viewing and language outcomes," *American Behavioral Scientist*, vol. 48, no. 5, pp. 624–645, 2005.

[109] D. Field and D. Anderson, "Instruction and modality effects on children's television attention and comprehension.," *Journal of Educational Psychology*, vol. 77, no. 1, pp. 91, 1985.

[110] L. Lam and C.Y. Suen, "Application of majority voting to pattern recognition: an analysis of its behavior and performance," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 27, no. 5, pp. 553–568, 1997.

[111] X. Sun, "Pitch determination and voice quality analysis using subharmonic-to-harmonic ratio," in *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*. IEEE, 2002, vol. 1, pp. I–333.

[112] B. Schölkopf, A. Smola, and K-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[113] M. Girolami, "Mercer kernel-based clustering in feature space," *Neural Networks, IEEE Transactions on*, vol. 13, no. 3, pp. 780–784, 2002.

[114] H.S.K.I. Noma, "Dynamic time-alignment kernel in support vector machine," *Advances in neural information processing systems*, vol. 14, pp. 921, 2002.

[115] Joseph Santarcangelo, "Dynamic-time-alignment-k-means-kernel-clustering-for-time-sequence-clustering," in *https://www.mathworks.com/matlabcentral/fileexchange/54358-jsantarc-dynamic-time-alignment-k-means-kernel-clustering-for-time-sequence-clustering*, 2015.

[116] J. Santarcangelo and X-P. Zhang, "Classifying harmful children's content using affective analysis," in *Multimedia Signal Processing (MMSP), 2014 IEEE 16th International Workshop on*. IEEE, 2014, pp. 1–6.

[117] H. Gunes, B. Schuller, M. Pantic, and R. Cowie, "Emotion representation, analysis and synthesis in continuous space: A survey," in *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*. IEEE, 2011, pp. 827–834.

[118] T.W. Liao, "Clustering of time series dataa survey," *Pattern recognition*, vol. 38, no. 11, pp. 1857–1874, 2005.

[119] S. Dalmiya, A. Dasgupta, and S.K. Datta, "Application of wavelet based k-means algorithm in mammogram segmentation," *International Journal of Computer Applications*, vol. 52, no. 15, pp. 15–19, 2012.

[120] M. Sjöberg, Y. Baveye, H. Wang, V.L. Quang, B. Ionescu, E. Dellandréa, M. Schedl, C-H. Demarty, and L. Chen, "The mediaeval 2015 affective impact of movies task," in *MediaEval 2015 Workshop*, 2015.

[121] M. Xu, L-T. Chia, and J. Jin, "Affective content analysis in comedy and horror videos by audio emotional event detection," in *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*. IEEE, 2005, pp. 4–pp.

[122] Y. Baveye, J-N. Bettinelli, E. Dellandréa, L. Chen, and C. Chamaret, "A large video data base for computational models of induced emotion," in *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on*. IEEE, 2013, pp. 13–18.

[123] Y. Baveye, E. Dellandrea, C. Chamaret, and L. Chen, "From crowdsourced rankings to affective ratings," in *Multimedia and Expo Workshops (ICMEW), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1–6.

[124] Y. Baveye, C. Chamaret, E. Dellandréa, and L. Chen, "A protocol for cross-validating large crowdsourced data: The case of the liris-accede affective video dataset," in *Proceedings of the 2014 International ACM Workshop on Crowdsourcing for Multimedia*. ACM, 2014, pp. 3–8.

[125] M.E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *The journal of machine learning research*, vol. 1, pp. 211–244, 2001.

[126] J. Shawe-Taylor and N. Cristianini, *Kernel methods for pattern analysis*, Cambridge university press, 2004.

[127] R.W. Lienhart, "Comparison of automatic shot boundary detection algorithms," in *Electronic Imaging'99*. International Society for Optics and Photonics, 1998, pp. 290–301.

[128] H. Leventhal and K. Scherer, "The relationship of emotion to cognition: A functional approach to a semantic controversy," *Cognition and emotion*, vol. 1, no. 1, pp. 3–28, 1987.

[129] R.D. Ray, "Emotion elicitation using films," *Handbook of emotion elicitation and assessment*, pp. 9–28, 2007.

[130] J.J. Gross and R.W. Levenson, "Emotion elicitation using films," *Cognition & emotion*, vol. 9, no. 1, pp. 87–108, 1995.

[131] A. Metallinou and S. Narayanan, "Annotation and processing of continuous emotional attributes: Challenges and opportunities," in *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*. IEEE, 2013, pp. 1–8.

[132] D. Hasler and S.E. Suesstrunk, "Measuring colorfulness in natural images," in *Electronic Imaging 2003*. International Society for Optics and Photonics, 2003, pp. 87–95.

[133] Y. Baveye, F. Urban, C. Chamaret, V. Demoulin, and P. Hellier, "Saliency-guided consistent color harmonization," in *Computational Color Imaging*, pp. 105–118. Springer, 2013.

[134] Y. Ke, X. Tang, and F. Jing, "The design of high-level features for photo quality assessment," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. IEEE, 2006, vol. 1, pp. 419–426.

[135] Y. Luo and X. Tang, "Photo and video quality evaluation: Focusing on the subject," in *Computer Vision–ECCV 2008*, pp. 386–399. Springer, 2008.

[136] T. Porter and T. Duff, "Compositing digital images," in *ACM Siggraph Computer Graphics*. ACM, 1984, vol. 18, pp. 253–259.

[137] Jingdong Wang, Jianguo Lee, and Changshui Zhang, "Kernel trick embedded Gaussian mixture model," in *International Conference on Algorithmic Learning Theory*. Springer, 2003, pp. 159–174.

[138] Sylvia Richardson and Peter J Green, "Bayesian analysis of mixtures with an unknown number of components (with discussion)," *Journal of the Royal Statistical Society: series B (statistical methodology)*, vol. 59, no. 4, pp. 731–792, 1997.

[139] Y. Bengio and P. Frasconi, "An input output HMM architecture," *Advances in neural information processing systems*, pp. 427–434, 1995.

[140] J. Wang, C. Xu, and E. Chng, "Automatic sports video genre classification using pseudo-2D-HMMs," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*. IEEE, 2006, vol. 4, pp. 778–781.

[141] H-B. Kang, "Affective content detection using HMMs," in *Proceedings of the eleventh ACM international conference on Multimedia*. ACM, 2003, pp. 259–262.

[142] X. Wang and X-P. Zhang, "Ice hockey shot event modeling with mixture hidden Markov model," in *Proceedings of the 1st ACM international workshop on Events in multimedia*. ACM, 2009, pp. 25–32.

[143] R. Cowie and M. Sawey, "Gtrace:www.psych.qub.ac.uk/gtrace," .

[144] J. Santarcangelo, "Kernel-based mixture of experts models for linear regression," *www.mathworks.com/matlabcentral/fileexchange/49064-kernel-based-mixture-of-experts-models-for-linear-regressionn*, 2015.

[145] Y. Bengio and P. Frasconi, "Input-output HMM's for sequence processing," *Neural Networks, IEEE Transactions on*, vol. 7, no. 5, pp. 1231–1249, 1996.

[146] J. Santarcangelo, "Dynamic prediction hidden Markov models in matlab," *http://www.mathworks.com/matlabcentral/fileexchange/46969-code-zip*, 2015.

[147] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[148] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, and G.E. Hinton, "Adaptive mixtures of local experts," *Neural computation*, vol. 3, no. 1, pp. 79–87, 1991.

[149] S.E. Yuksel, J.N. Wilson, and P.D. Gader, "Twenty years of mixture of experts," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 23, no. 8, pp. 1177–1193, 2012.

[150] M. Szummer and R.W. Picard, "Temporal texture modeling," in *Image Processing, 1996. Proceedings., International Conference on*. IEEE, 1996, vol. 3, pp. 823–826.

[151] K. Laybourne, *The animation book: a complete guide to animated filmmaking–from flip-books to sound cartoons to 3-D animation*, Three Rivers Press, 1998.

[152] R.M. Haralick, K. Shanmugam, and I.H. Dinstein, "Textural features for image classification," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 3, no. 6, pp. 610–621, 1973.

[153] J. Zhang and T. Tan, "Brief review of invariant texture analysis methods," *Pattern recognition*, vol. 35, no. 3, pp. 735–747, 2002.

[154] D.A. Clausi, "An analysis of co-occurrence texture statistics as a function of grey level quantization," *Canadian Journal of remote sensing*, vol. 28, no. 1, pp. 45–62, 2002.

[155] S.W. Zucker and D. Terzopoulos, "Finding structure in co-occurrence matrices for texture analysis," *Computer graphics and image processing*, vol. 12, no. 3, pp. 286–308, 1980.

[156] L.K. Soh and C. Tsatsoulis, "Texture analysis of sar sea ice imagery using gray level co-occurrence matrices," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 37, no. 2, pp. 780–795, 1999.

[157] L. Guan, Y. Wang, R. Zhang, Y. Tie, A. Bulzacki, and M. Ibrahim, "Multimodal information fusion for selected multimedia applications," *International Journal of Multimedia Intelligence and Security*, vol. 1, no. 1, pp. 5–32, 2010.

[158] M. Wang, X.S. Hua, J. Tang, and R. Hong, "Beyond distance measurement: Constructing neighborhood similarity for video annotation," *Multimedia, IEEE Transactions on*, vol. 11, no. 3, pp. 465–476, 2009.

[159] J. Santarcangelo, X-P. Zhang, and A. Wu, "An optimal automated signal superimposing method for remote field eddy current transformer coupling signals," in *Emerging Signal Processing Applications (ESPA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 107–110.

chapter